N° d'ordre : 4678

THÈSE

PRÉSENTÉE A

L'UNIVERSITÉ BORDEAUX 1

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

Par Willy AUBRY

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Etude et mise en place d'une plateforme d'adaptation multiservice embarquée pour la gestion de flux multimédia à différents niveaux logiciels et matériels

Soutenue le : 19/12/2012
Devant la commision d'examen formée de :

Mr.	Andre-Luc Beylot	Professeur, IRIT, Toulouse III	President de Jury
Mr.	Zoubir Mammeri	Professeur, IRIT, Toulouse III	Rapporteur
Mr.	Chiheb Rebai	Professeur, Ecole SUP'COM Tunis	Rapporteur
Mme.	Francine Krief	Professeur, LaBRI, Bordeaux I	Co-directrice de thèse
Mr.	Dominique Dallet	Professeur, IMS, Bordeaux I	Co-directeur de thèse
Mr.	Bertrand Le Gal	Maitre de Conférence, IMS, Bordeaux I	Co-encadrant de thèse
Mr.	Daniel Negru	Maitre de Conférence, LaBRI, Bordeaux I	Co-encadrant de thèse
Mr.	Sebastien Tatin	Ingénieur, Viotech Communications	Examinateur

Resumé

Les avancées technologiques ont permis la commercialisation à grande échelle de terminaux mobiles. De ce fait, l'homme est de plus en plus connecté et de partout. Ce nombre grandissant d'utilisateurs du réseau ainsi que la forte croissance du contenu disponible, aussi bien d'un point de vue quantitatif que qualitatif saturent les réseaux et l'augmentation des moyens matériels (passage à la fibre optique) ne suffisent pas. Pour surmonter cela, les réseaux doivent prendre en compte le type de contenu (texte, vidéo, ...) ainsi que le contexte d'utilisation (état du réseau, capacité du terminal, ...) pour assurer une qualité d'expérience optimale. A ce sujet, la vidéo fait partie des contenus les plus critiques. Ce type de contenu est non seulement de plus en plus consommé par les utilisateurs mais est aussi l'un des plus contraignants en terme de ressources nécéssaires à sa distribution (taille serveur, bande passante, …). Adapter un contenu vidéo en fonction de l'état du réseau (ajuster son débit binaire à la bande passante) ou des capacités du terminal (s'assurer que le codec soit nativement supporté) est indispensable. Néanmoins, l'adaptation vidéo est un processus qui nécéssite beaucoup de ressources. Cela est antinomique à son utilisation à grande echelle dans les appareils à bas coûts qui constituent aujourd'hui une grande part dans l'ossature du réseau Internet.

Cette thèse se concentre sur la conception d'un système d'adaptation vidéo à bas coût et temps réel qui prendrait place dans ces réseaux du futur. Après une analyse du contexte, un système d'adaptation générique est proposé et évalué en comparaison de l'état de l'art. Ce système est implémenté sur un FPGA afin d'assurer les performances (temps-réel) et la nécessité d'une solution à bas coût. Enfin, une étude sur les effets indirects de l'adaptation vidéo est menée.

Mots clefs : Vidéo, Transcodage, FPGA, Réseau Domestique

Résumés des travaux et résultats

Les avancées technologiques ont permis la commercialisation à grande échelle de terminaux mobiles. De ce fait, l'homme est de plus en plus connecté et de partout. Ce nombre grandissant d'utilisateurs du réseau ainsi que la forte croissance du contenu disponible, aussi bien d'un point de vue quantitatif que qualitatif saturent les réseaux. L'augmentation des moyens matériels (passage à la fibre optique) ne suffisent pas. Les mécaniques des réseaux doivent prendre en compte le type de contenu (texte, vidéo, ...) ainsi que le contexte d'utilisation (état du réseau, capacité du terminal, ...) pour assurer une qualité d'expérience optimale. A ce sujet, la vidéo fait partie des contenus les plus critiques. Ce type de contenu est non seulement de plus en plus consommé par les utilisateurs mais est aussi de plus en plus produit par l'utilisateur. En outre, le contenu vidéo est un des contenu les plus contraignants que ce soit en terme de ressources nécéssaires à sa distribution (taille serveur, bande passante, …) ou en nombre de techniques de distribution (UNICAST, MULTICAST, …). Le transport d'un contenu vidéo adapté à l'état du réseau (ajuster son débit binaire à la bande passante) et aux capacités du terminal (s'assurer que le codec et la résolution soient nativement supportés) est indispensable. Afin de répondre à cette nécéssité, plusieurs solutions ont été envisagées : le sur-approvisionnement de réseaux/données, l'encodage par couche et l'adaptation vidéo.

Le chapitre 1 décrit les contraintes de cette distribution de flux vidéo et analyse ces solutions afin de comparer leur capacité de réponse à ce nouveau contexte. L'adaptation vidéo est établie comme la meilleure solution pour une nouvelle génération du réseau centrée sur la consommation de média. Elle conserve les optimisations qu'apportent les différentes techniques de distribution et permet à l'utilisateur de fournir son propre contenu sans devoir passer par une tierce partie. La réponse de manière efficace aux nouvelles utilisations du réseau nécéssite de positionner cette adaptation au plus près de l'utilisateur. Dans un contexte de réseau à domicile, cette localisation est la passerelle domestique, centre du réseau de la maison et interface unique avec le réseau extérieur. Néanmoins, cette localisation pose un problème. Dans le but de garantir une qualité d'expérience optimale, le processus d'adaptation doit être temps réel. Cette contrainte de temps réel est respectée si le processus d'adaptation a une cadence plus élevée que le nombre d'images par seconde du flux vidéo. Cela nécéssite un nombre de ressources de calcul important. Cette demande de puissance est antinomique à son utilisation à grande echelle dans les passerelles domestiques qui sont des appareils à bas coûts. Non seulement un processus d'adaptation moins coûteux doit être recherché mais son implémentation doit être faite sur accélérateur matériel, composant plus efficace que les processeurs généralistes.

Le chapitre 2 présente le processus d'adaptation et décrit les travaux effectués dans ce domaine. Le processus d'adaptation est composé de 3 étapes : (1) le décodage du flux entrant, (2) la modification des données décodées et (3) l'encodage des données modifiées. Ces trois étapes permettent d'effectuer toutes les adaptation vidéos répertoriées que sont (1) le changement de résolution spatiale et/ou (2) temporelle, (3) le changement de débit binaire, (4) le changement de codec ou (5) une combinaison de deux ou plus des modifications sus-citées. Les travaux effectués dans le domaine de l'adapation vidéo ont pour objet principal, la réduction de la complexité calculatoire d'un des processus d'adaptation tout en maintenant une qualité vidéo maximale. Toutes ces contributions proposent de réduire la complexité de calcul du processus en exploitant les informations contenues dans le flux entrant (et donc décoder par la partie « décodage ») afin de simplifier la partie « encodage », notamment grâce à l'utilisation d'heuristiques. Par exemple, les vecteurs de mouvement peuvent être réutilisés lors de l'encodage afin de réduire le temps de calcul lors de l'étape d'estimation de mouvement, voire de supprimer l'étape complètement. Il est donc important de comprendre les mécanisme d'encodage et de décodage. Ces mécanismes sont présentés dans la première partie du chapitre 2 en amont de l'état de l'art dans le domaine de l'adaptation vidéo.

Une vidéo est considérée comme une succession d'images dans le temps. Les mécanismes de compression et décompression vidéo empruntent donc une partie de leur concept des mécanismes utilisés lors de la compression/décompression d'image. Ainsi pour la compression, une image est représentée dans un domaine colorimétrique (YUV pour la vidéo) et est découpée en macroblocs. Un macrobloc étant un ensemble de blocs sélectionné dans les différents plans colorimétriques. Par exemple, un macrobloc MPEG-2 correspond, pour le domaine YUV 4:2:0, à un regroupement de 4 blocs de 8x8 pixels adjacents dans le plan de luminance et à 2 blocs de 8x8 pixels sélectionnés respectivement dans les deux plans de chrominances (U et V). Une transformée en fréquence (par exemple : transformée en cosinus discrète) est alors opérée sur les macroblocs. Ces nouveaux coefficients sont alors quantifiés par un scalaire dépendant de la fréquence qu'il représente. En effet, l'oeil humain est moins sensible aux hautes fréquences (détails) qu'aux basse fréquences. Le facteur de quantification est alors plus grand (plus de pertes) pour les hautes fréquences que les basses. Enfin une succession d'encodages à longueur variable (comme le « Run Length Coding » et/ou le « Variable Length Coding ») sont utilisés pour terminer la compression de l'image.

Mais le processus d'encodage vidéo comprend aussi des mécanismes qui lui sont propres, tel que la prédiction temporelle. En effet, les informations présentes dans une image sont souvent présentes dans l'image précédente quelle soit au même endroit ou qu'elles aient bougées (vecteur de mouvement). Il est donc possible de diminuer les informations contenues dans l'image en utilisant les informations contenues dans l'image précédente et donc en recherchant les vecteurs de mouvement qui décrivent le passage d'une image à l'autre. Les macroblocs utilisant ces mécanismes sont dits de type INTER.

Le processus de décodage procède à l'inverse du processus d'encodage. Il opère un décodage entropique pour restituer les coefficients qu'il multiplie par un scalaire puis effectue une transformée inverse pour revenir dans le domaine spatial. Enfin, si le type du macrobloc décodé est INTER, le processus de décodage utilise les vecteurs de mouvement contenus dans le flux pour ajouter aux macrobloc les données contenues dans l'image précédente.

Les propositions de simplification du processus d'adaptation adressent des contextes précis et différents. Par exemple, une adaptation qui ne diminue que le bitrate mais pas les autres paramètres (résolutions spatial/temporelle ou codec) est proposée pour réduire l'impact de la vidéo sur la bande passante du réseau. Dans ce contexte, des processus ont été présentés opérant dans le domaine des fréquences (i.e. en supprimant la transformée dans les étapes d'encodage et de décodage) afin d'éviter les étapes de transformée et transformée inverse. Le chapitre 2 présente ces différentes propositions qui visent principalement à supprimer des étapes lors des processus de décodage et d'encodage. Nous établissons la conclusion que chacun des processus d'adaptation présentés résoud un sous ensemble d'adaptations précises en fonction du contexte dans lequel ils se situent. Il n'existe pas d'étude connue axée sur l'utilisation d'un processus utilisé pour une adaptation précise (ex. changement de résolution) afin d'en réaliser une autre (ex. changement de bitrate). Aucune proposition connue de processus d'adaptation se situe dans un contexte généraliste où l'ensemble des adaptations serait demandé. Par conséquent, il n'existe pas de processus d'adaptation connu qui résout le problème d'une adaptation généraliste répondant à notre problématique de faible coût et de temps réel.

Les règles permettant de répondre à la définition d'une adaptation dite « générique » sont élaborées lors du chapitre 3 et la sélection d'un sous-ensemble de simplifications identifiées au chapitre 2 et répondant à ces règles est conduite. En définissant le processus d'adaptation générique comme un processus capable d'opérer une combinaison d'un ou de plusieurs des quatres types d'adaptations vidéo sur un flux vidéo encodé par un codec quelconque, ce processus ne doit pas dépendre de la transformée (en cosinus discrète, entière, …) utilisé par les codecs des vidéos originales ou finales. Ainsi la partie « modification des données » doit opérer dans le domaine spatial représentant les pixels. D'autre part, la possibilité d'avoir, dans une même image, la possibilité d'encoder certains pixels avec ou sans une référence à d'autres pixels présents dans des images déjà décodées (voir chapitre 2) contraint la partie « décodage » à ne pas supprimer entièrement l'étape

ultime de reconstruction par compensation de mouvement. Ces deux règles interdisent l'utilisation de certaines techniques afin de simplifier le processus d'adaptation vidéo.

Enfin, en seconde partie du chapitre 3 sont faite la proposition et l'évaluation de deux processus d'adaptation vidéo répondant aux contraintes de l'adaptation dans les réseaux : la « generic partial encode » et la « generic intra refresh ». La proposition de ces deux processus repose, d'une part, sur l'étude des techniques de simplification menée précédement et, d'autre part, sur les règles édictées afin d'obtenir un processus d'adaptation générique. L'évaluation de qualité est menée par (1) la sélection de métriques de qualité et (2) l'élaboration d'un banc de test. La métrique utilisée doit reflèter la qualité perçue. Ainsi le « Structural SIMilarities » (SSIM) est utilisé. Le banc de test doit reflèter les conditions réelles de consommation vidéo afin de ne pas apporter de biais. Les cas d'utilisation de l'adaptation vidéo sont donc décrits et utilisés pour l'élaboration du banc de test. Trois processus sont alors évalués : la « generic partial encode », la « generic intra refresh » (les deux processus proposés) ainsi que le processus de référence (opération d'adaptation sans simplification). Les résultats sont alors fournis par les mesures de SSIM et de PSNR. Ces résultats sont ensuite mis en perspectives par rapport aux gains apportés par les simplifications oppérées sur le processus d'adaptation. Ces résultats permettent de conclure qu'une des propositions (« generic intra refresh ») n'atteint pas une qualité de vidéo suffisante pour permettre de la retenir. En effet, bien que son SSIM moyen soit compris entre 70% et 92%, la variation temporelle de SSIM est trop grande (ecart type entre 1.8% et 5.2%). En revanche, la seconde proposition (« generic partial encode ») atteint une qualité remarquable (SSIM compris entre 96% et 99% en fonction des vidéos ; écart-type dans le temps inférieur à 0.05%)pour une complexité moindre au regard du processus de référence et répond donc à notre problématique.

Le chapitre 3 a permis l'élaboration d'un processus d'adaptation à complexité moindre. Le chapitre 4 se concentre sur les architectures matérielles d'intégration matérielle de ce processus. Ces architectures doivent tenir compte du caractère générique du procédé tout en essayant d'être le moins coûteux (ressource silicium) possible. Il convient alors de trouver une architecture qui soit plus performante que l'adjonction côte à côte de toutes les possibilités d'adaptation. Il est alors proposé de réutiliser les circuits d'encodeurs et de décodeurs mais également le circuit de modification de données (celui placé entre le décodeur et l'encodeur dans le processus d'adaptation) qui doit donc être générique. Cette généricité nécéssite un mécanisme de sélection de l'adaptation (résolution spatiale/temporelle, bitrate) qui est détaillé dans le début du chapitre. Dans le cas d'une adaptation multiple (i.e. bitrate et résolution spatiale), il est important de définir l'ordre des procédés de modification (i.e. bitrate en premier puis résolution ou l'inverse) afin d'optimiser le nombre de calculs. Nous avons conclu que la modification de la résolution temporelle doit être effectuée en premier puis la modification de la résolution spatiale et enfin le contrôle du bitrate.

Enfin, le circuit de modification de données pour être utilisable par tous les encodeurs et décodeurs doit opérer dans un format commun à tous. La seconde partie du chapitre 4 présente les problématiques liées à l'élaboration de ce format commun et propose un format d'adaptation répondant à ces problématiques. Ce format doit pouvoir inclure toutes les spécificités des CODEC présents. Il opère dans le domaine spatial (pixels) et contient un espace de métadonnées qui est obtenu par l'union des espaces des métadonnées des différents CODEC. L'ordre des données est également important. Pour un traitement optimal nous proposons le regroupement des données en sous blocs de 4x4 pixels afin d'exploiter la granularité la plus fine. L'implémentation d'un changement de format du standard MPEG-2 au format d'adaptation commun et réciproquement sont décrits.

Utilisant les résultats et conclusions des deux premières parties du chapitre 4, la troisième partie se concentre sur l'architecture globale du processus d'adaptation. Les différentes adaptations doivent être possibles quelque soit le codec. Deux architectures sont proposées. La première dite statique utilise une interconnexion par bus afin de piloter le routage des données suivant les codeur/décodeur/adaptation appropriés. La seconde architecture exploite les capacités de reconfiguration dynamique et partielle des puces électroniques connues sous le nom de FPGA (Field

Programmable Gate Array). Cette reconfiguration dynamique permet de réduire drastiquement la taille de la puce en ne configurant le FPGA qu'avec le circuit correspondant à la demande d'adaptation. Nos résultats d'implémentation d'un transcodeur générique configuré pour faire un transcodage MPEG-2 vers MPEG-2 (que nous avons développé suivant les résultats du chapitre 3 et 4) demandent 3% de slices en moins, 54% de BRAM en moins et 11% de multipliers en moins, en comparaison avec des résultats d'implémentations industriels.

A l'issue du chapitre 4, nous avons obtenu une implémentation fonctionnelle d'un processus d'adaptation qui est générique, moins coûteux en sillicium et temps réel tout en conservant une haute qualité de vidéo. Cette implémentation est utilisée pour évaluer les impacts indirects de l'adaptation vidéo sur l'expérience utilisateur. Cette évaluation est décrite dans le dernier chapitre de cette thèse. Diminuer la résolution spatiale d'une vidéo est utilisée principalement pour permettre aux terminaux de consommer des vidéos qui ont une résolution originale trop élevée. Mais la diminution de la résolution spatiale a d'autres effets tout aussi importants. D'une part, la réduction du nombre de pixels dans chaque image réduit le débit binaire de la vidéo. Cela permet de résoudre des problèmatiques de bande passante et de congestion réseau mais également de réduire la consommation du terminal en réception (moins de données). D'autre part, et toujours dans une optique de consommation d'énergie, réduire le nombre de pixels d'une vidéo réduit le nombre de pixels à décoder et donc l'énergie nécessaire au décodage de la vidéo.

Deux cas se présentent alors : (1) la résolution spatiale de la vidéo originale est inférieure ou égale à la résolution de l'écran du terminal ou (2) la résolution spatiale de la vidéo originale est supérieure à la résolution de l'écran du terminal. Dans le premier cas, il n'y aurait pas besoin d'adaptation au premier abord. Cependant, le coût en énergie de la mise à l'échelle (agrandissement) de la vidéo est à mettre en relation avec le gain d'énergie apporté par la diminution du nombre de pixels. Dans le second cas, s'il n'y a pas adaptation, c'est le terminal qui va décoder la vidéo et qui va ajuster la résolution spatiale de la vidéo à la résolution d'affichage. Le terminal va donc consommer de l'énergie en trop pour décoder et ajuster le surplus de pixels par rapport à la réception d'une vidéo déjà à l'échelle. Nos résultats montrent qu'une réduction d'énergie d'environ 50% est obtenue pour une division par deux des dimensions spatiales de la vidéo et que le gain varie entre 70% et 90% pour une division par quatre des dimensions spatiales de la vidéo. Les qualités vidéos sont les mêmes que celle évaluées lors du chapitre 3 (90% de ressemblance avec l'original).

Cette thèse présente donc une étude sur la consommation vidéo dans les réseaux et l'émergence de l'adaptation vidéo comme un des processus clefs de l'évolution du réseau vers une utilisation plus intelligente et plus performante. Cette adaptation doit cependant être temps réel et à faible coût. Une présentation succincte des techniques de décodage et d'encodage vidéo est faite afin d'aborder l'état de l'art en matière de processus d'adaptation. De cet état de l'art est tirée la conclusion qu'il n'existe pas de processus d'adaptation connue qui respecte nos contraintes. Une analyse est alors conduite et deux propositions de processus d'adaptation sont faites. Ces propositions sont évaluées. Une fois, le processus sélectionné, les problématiques d'implémentation sur accélérateur matérielle (FPGA) d'un tel processus sont soulevés et des propositions sont fournies notamment à travers l'utilisation d'un format d'adaptation universel et l'utilisation de la reconfiguration dynamique. Les coûts d'un accélérateur matériel pour une adapation vidéo générique et temps réel sont drastiquement réduits. Ainsi, l'implémentation d'un tel acclérateur ayant été réalisé, une utilisation de l'adaptation vidéo pour réduire la consommation d'énergie au niveau du terminal utilisateur est proposée et évaluée. En conclusion de cette thèse, sont d'abord présentées les problématiques à venir telle l'arrivée de nouveaux standards vidéo (HEVC) puis des axes de recherche sont donnés aussi bien pour optimiser l'utilisation de la reconfiguration dynamique (traitements en parrallèle) que pour détecter le besoin d'adaptation et la sélection de l'adaptation la plus appropriée au besoin.

Title : Conception and implementation of an hardware accelerated video adaptation platform in a home network context

Abstract :

On the one hand, technology advances have led to the expansion of the handheld devices market. Thanks to this expansion, people are more and more connected and more and more data are exchanged over the Internet. On the other hand, this huge amound of data imposes drastic constrains in order to achieve sufficient quality. The Internet is now showing its limits to assure such quality. To answer nowadays limitations, a next generation Internet is envisioned. This new network takes into account the content nature (video, audio, ...) and the context (network state, terminal capabilities ...) to better manage its own resources. To this extend, video manipulation is one of the key concept that is highlighted in this arising context. Video content is more and more consumed and at the same time requires more and more resources. Adapting videos to the network state (reducing its bitrate to match available bandwidth) or to the terminal capabilities (screen size, supported codecs, …) appears mandatory and is foreseen to take place in real time in networking devices such as home gateways. However, video adaptation is a resource intensive task and must be implemented using hardware accelerators to meet the desired low cost and real time constraints.

In this thesis, content- and context-awareness is first analyzed to be considered at the network side. Secondly, a generic low cost video adaptation system is proposed and compared to existing solutions as a trade-off between system complexity and quality. Then, hardware conception is tackled as this system is implemented in an FPGA based architecture. Finally, this system is used to evaluate the indirect effects of video adaptation; energy consumption reduction is achieved at the terminal side by reducing video characteristics thus permitting an increased user experience for End-Users.

Key Words : Video, Transcoding, FPGA, Home Network

Acknowledgments

Je remercie Francine Krief et Dominique Dallet pour m'avoir intégré dans leur équipe et d'avoir dirigé mes recheches par leur relectures et invitations à participer à différents événements de la vie de laboratoire.

Je remercie Daniel Negru pour son encadrement et son implications dans les projets, notemment pour sa volonté de me faire participer au projet ANR ARDMAHN .

Je remercie Bertrand Le Gal, pour son encadrement, son énergie et son amitié. J'espère pouvoir continuer de collaborer avec lui dans le foisonnement d'idées qui lui est toute caractéristique.

Je remercie Zoubir Mameri et Chiheb Rebai pour le temps qu'ils ont pris à l'édification d'un rapport sur mes travaux. Je les remercie ainsi qu'Andre-Luc Beylot et Sebastien Tatin de leur venue à Bordeaux pour assister à ma soutenance et juger de l'interet de mes travaux dans le cadre de l'obtention du titre de Docteur. Je sais maintenant que mes travaux ont été dignes du temps qu'ils y ont consacrés.

Je remercie toute l'équipe de Viotech Communications pour les moyens mis en oeuvre et l'aide apportée tout au long de cette thèse, notemment la confrontation aux considération du client. Je tiens à remercier en particulier julien Pauty pour ses premiers conseils de direction de thèse, ainsi que Pete Sedcole et Sebastien Tatin pour avoir partager leur expériences avec moi.

Je remercie Simon Desfarges pour son travail mais aussi sa complicité et ses débats au jour le jour qui ont permis la remise en cause de certaines décisions et l'aboutissement (en collaboration avec Viotech Communications) d'un prototype fonctionnel.

Je voudrais finir ces remerciements en remerciant mon épouse Aline, ma famille que ce soit du coté Aubry ou du Mas des Bourboux, ainsi que mes amis pour leur soutien moral, leurs sourires et leur acceuil.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

List of Publications

International Journals

F. Duhem, F. Muller, W. Aubry, B. Le Gal, D. Négru and P. Lorenzini, “Design Space Exploration for Partially Reconfigurable Architectures in Real-Time Systems”, **under review** at Journal of Systems Architecture, submited in May 2012

International Conferences

W. Aubry, B. Le Gal, D. Negru, S. Desfarges and D. Dallet, “A Generic Video Adaptation FPGA Implementation towards Content – and Content – Awareness in Future Network”, IEEE International Conference on Design and Architecture for Signal and Image Processing (DASIP), October 2012

W. Aubry, D. Negru, B. Le Gal, S. Desfarges and D. Dallet, “A Generic Video Adaptation Framework Towards Content - and Context - Awareness in Future Networks”, IEEE European Signal Processing Conference (EUSIPCO), August 2012, pages 2218-2222

W. Aubry, D. Negru, B. Le Gal and D. Dallet, “Spatial Downsizing Impact in the Transrating Tradeoff for content/context Awareness in Media Networks”, IEEE International Conference on Telecommunication and Multimedia (TeMU), July 2012, pages 158-162

W. Aubry, B. Le Gal, D. Negru, D. Dallet and S. Desfarges, “A system approach for reducing power consumption of multimedia devices with a low QoE impact”, IEEE Conference on Electronics Circuit and Systems (ICECS), December 2011, pages 5-8

E. Casseau, S. Khan, B. Le Gal, and W. Aubry, “Multimode architecture design”, IEEE International Conference on Design and Architectures for Signal and Image Processing, (DASIP), November 2007

French Conferences

W. Aubry, D. Negru and P. Kadionik, “Video Adaptation Acceleration in a Home Networking context”, GDR SOC-SIP, June 2009

Award

The SFR-Fondation Bordeaux Université award obtained on the 13th December 2012

ARDMAHN Project Deliverables

D.1.1: “Analyse des Cas d’Utilisation et Spécification de la Home Gateway”, http://ardmahn.org/documents/d11.pdf

D.3.0 : “Analyse et choix des algorithmes d’adaptation entre standards de compression”, ardmahn.org/documents/d30.pdf

Chapter 0 : Introduction

In the last decade, the Internet has become a revolution for our way of life. Reading, watching, learning, playing, buying, socializing, everything can be done on the Internet. Information can be generated and relayed by anyone with an Internet connection and is no more reserved to specialized parties. Nearly a third of Earth's human population is consuming/generating content on/for the Internet.

With upgraded capabilities, embedded devices and more precisely handheld ones are now able to be constantly connected, leading to a continuous use of Internet applications on them. The user is now able to access the Internet while moving from a place to another, taking pictures/videos from anywhere and instantly making them available to anyone.

Thanks to this sky rocketing amount of users and contents, it is very likely that the Internet will soon reach an overloaded state. Due to the deployment costs and infrastructure needs, even with major technology advances, such as in optic fiber, we might not be able to cope with this huge amount of traffic if we stay with the current approach. These issues have been perceived and analyzed in [FIA11]. Solutions are being proposed at international scale along with strong research initiatives in the 'Future Internet' fields, especially related to media delivery. At the European scale, the FP7 ALICANTE project [ALI] proposes a new media ecosystem that overlays the global network with intelligent routing and media adaptation. At the French level, the ARDMAHN project [ARD] envisions an auto configurable media gateway that has video adaptation capabilities. These projects foresee moving from a content agnostic Internet to a content- and context-aware network for the Future Internet.

To this case, video content is definitely foreseen as one of the most critical content. On the one hand, it is one of the most consumed media on the Internet. On the other hand, it is one of the most resources requiring in terms of bandwidth and server space.

Video content can be declined to various quality levels depending on frame quality, frame resolution and frame rate. However, the quality of the viewing experience is not only based on the video quality but is also constrained by the accessibility of the video (e.g. having the proper codec) or the lag during this viewing experience (induced by insufficient bitrate).

We study the possibility to enhance content and context aware network by allowing them to adapt video content if needed. As a result, video content may be adapted considering network or end user's terminal context. Video adaptation enables: (1) content access through codec change (from an unsupported codec to a codec supported by the end user's terminal) and (2) improved quality of experience through the reduction of video bitrate to the available bandwidth.

The work presented within this thesis has been performed in order to provide a solution to the need to adapt the video quality level in order to optimize the quality of experience of the End-Users. We analyse video adaptation in content and context aware network. Thanks to this analysis, we propose a generic and real time video adaptation system that operate in a enhanced and low cost home gateway. Toward these constraint, state of the art video adaptation algorithms are overviewed, new generic video adaptation are proposed and evaluated and solutions to hardware acceleration of such system are proposed.

This thesis has been written in five chapters that highlight our working steps. First and foremost, ***chapter 1*** details and analyzes the overall context of the study. From this analysis, all constraints needed to be fulfilled by our video adaptation system are identified. Next, in ***chapter 2***, a comprehensive state of the art on video compression and decompression techniques is done. These

techniques are essential to fully understand the video adaptation systems proposed in the literature and overviewed in this chapter, as well.

A deep analysis of the abovementioned adaptation systems in accordance with the context is conducted in ***chapter 3***. From this analysis, their weaknesses are identified and our own video adaptation system is proposed, evaluated and compared to those. This system that proposes the best tradeoff between costs (measured in computational complexity) and quality (measured in the quality of adapted video) will be specified in order to be implemented.

Indeed, ***chapter 4*** details the implementation framework of the proposed video adaptation system. It highlights the key features and the important development costs reduction this multi-codec adaptation system can bring. Implementation results are provided to conclude this chapter.

Chapter 5 goes a step further and studies the use of our system to solve indirect effects of video adaptation, such as energy saving at the terminal side for a reduced resolution adaptation.

We conclude by outlining the key results of the thesis work and by describing the exploring areas that would need further focus in order to optimize the use of the proposed system.

Chapter 1 : Context and associated issues

1 Towards a Future Internet

Nowadays, the Internet is widely deployed and restlessly expanding all over the world. Hyperlink, handshake, error correction, packet routing …, the premises of data exchange have been set. We have created various network protocols that aim at delivering a particular type of data through a particular type of network (wired or wireless). From text messages, we are now able to access dynamic content and to interact with it. The technical advances in the academic and industrial worlds have led to new ways of using the network. Every day life is becoming more and more connected. Chatting, learning, buying, playing, the Internet is being extensively used.

1.1 New connected devices

Thanks to advances in microelectronics, devices embed a lot more computing resources than ever. A few years ago, only desktop computers were able to connect to the Internet. By adding these new features, the previously known cellular phones and PDAs have merged to become the “smart” phones (now called “smartphones”). As shown in Figure 1- 1, the smartphone market is exploding, leading to outperform the PC market. In addition, the PC world is experiencing another revolution along with the mobility feature, around the development of first laptops, then netbooks, ultrabooks and now the tablets. Tablets are more handheld than laptops, besides having wider screens than smartphones and are thus well suited and pleasant for consuming media on the go.

All these new devices have created a new population of mobile users that access data while moving from a place to another. The ubiquitous feature of the Internet is prevalent.

1.2 A new content consumption

1.2.1 A new type of user

Seizing their new condition of mobile users, people want to access data on where to go or what to find near the place they stand. They also want to provide multimedia content to their friend on the place they are, or on what they are doing now. The time when people were used to take pictures with their camera and show it to their friends by organizing a “picture showing party” when they return from their trip is over. The trend is to provide their personal virtual space on the Internet with photo or video contents for their friends to watch. The user is slowly adding a provider notion to his consumer position, becoming what can be named a “prosumer” (provider+consumer).

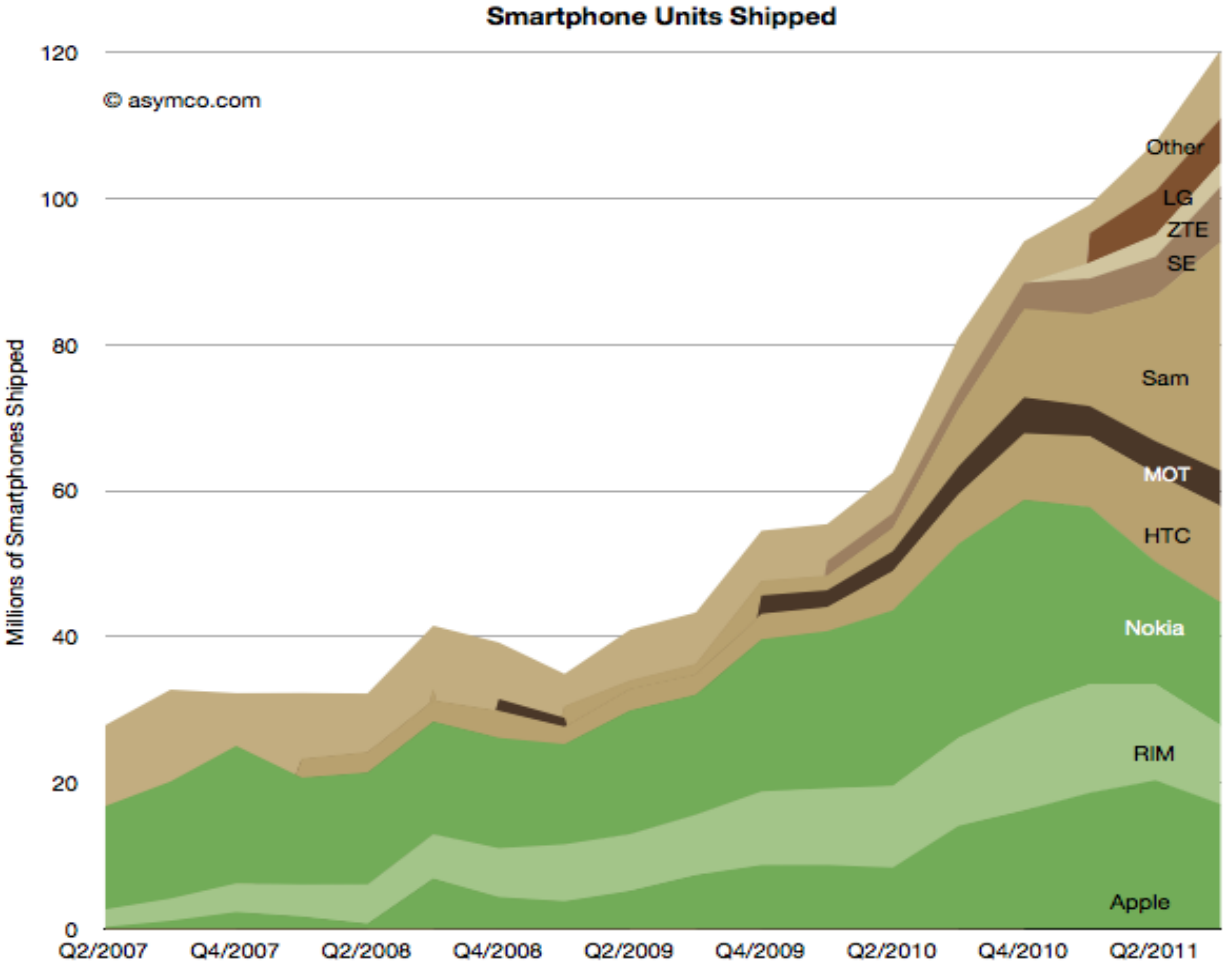


Figure 1- 1 : Smartphone Market Evolution[1]

The Internet is flooded with provider- and now more and more with also user-generated content ubiquitously offered over a variety of applications and in tight integration with social media platforms. There is an increasing trend towards live communications (in text, voice and video), audio/video consumption and user-generated content publication and consumption.

1.2.2 An evolving Networked Media ecosystem

From a consumer point of view, as depicted in Figure 1- 2, today's networked media ecosystems are made of several partners that collaborate to provide and deliver content to the user which consumes it. On the one hand, the Content Creator creates content that is published by Content/Service providers. Content/Service providers propose different kinds of content/service offers linked to the content they possess. On the other hand, end users receive and consume the contents/services. In the middle, the network provider is needed for transporting every data between the content/service provider and the consumer.

[1] http://www.asymco.com/2011/11/17/the-global-smartphone-market-landscape/

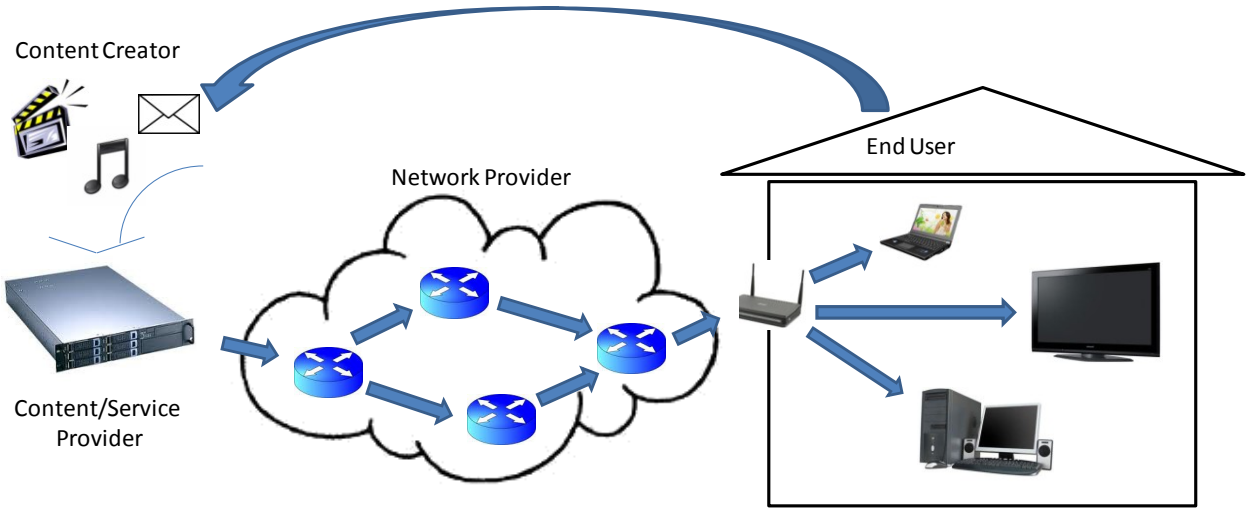


Figure 1- 2 : Classic Multimedia Delivery Chain

But the Internet shows its limit as end users also become content creator. The whole network is flooded by user generated content and the various actors in the media delivery chain are impacted.

Content creator: We will distinguish the professional Content Creators (authors, musicians, music/movie companies) from non-professional ones (amateur individuals). Profesionnals are linked with content providers that posess dedicated platforms. However, those platforms are professional oriented and, besides being expensive, do not facilitate the creation, publication, sharing and other features of non-professionnal user generated content. Therefore, non-professional Content Creators willing to act as prosumers by serving their own content through the network, use centralised content aggregator platforms (such as YouTube, live webcast or social sites) that not only offer limited control over one's content but also take one's content ownership.

Content Provider: Content Providers want their media content to reach maximum audience. Hence, they aim at distributing it with the greatest quality and the highest flexibility. Today, HD and 3D video are becoming commonplace in distributed content for TV-display devices. Resolutions are foreseen to progress to Ultra-HD in 2, 4 and 8K, featuring 3D and interactivity. Thus, Content Providers will require even higher bandwidth, low latency and low loss.

Today, it is mandatory for the End-Users to use a specific device to watch streamed content (e.g. IPTV stream with a specific set-top-box) or to choose between various proposed formats for consumption on their various terminals. For this purpose, Content Providers seek unified, simple tools to create new types of content from various Content Creators for ubiquitous consumption, while keeping the cost of distribution low.

Network Provider: The Network Providers have the difficult role of transporting content, data traffic having large diversity. In order to provide an acceptable degree of network-level Quality of Service, Network Providers apply resource management strategies such as i) Over-provisioning; i.e. accommodating more resources than needed and serving the traffic either in a best effort mode or applying coarse traffic differentiation to traffic aggregates, or ii) Static Provisioning of a given amount of resources. In any case, the network usually functions in a content-agnostic mode, i.e. applies the same treatment to all flows, regardless of the application which they convey and the requirements of each application.

End user: The End-User is seen as the last element in the Networked Media chain. The End-Users have always been Content Consumers and from recently are becoming Content Creators as well, especially thanks to user-participating and social applications within the Web 2.0/3.0 paradigm.

However, the media distribution model still relies on deployed servers, periodically re-dimensioned and replicated to serve the growing mass of potential consumers. As stated previously, the End-Users are experiencing limited interaction and flexibility with regard to creation, distribution and (mostly) management of their own content.

In addition to the limitations they face while acting as Content Creators, End-Users are still experiencing limitations as Content Consumers in their own environment. The proliferation of heterogeneous devices (tablets, smartphones, laptops, TVs, set-top-box) with various capabilities for service access and content display is leading to the necessity of providing the same content and services via all terminals, with the best possible quality, while at home or away. Today's deployed solutions, such as the ones proposed by telecommunications and mobile operators (web- and mobile portals, home gateways, user profiling, terminals' monitoring agents) still do not fulfil the needs for Media Service/Content ubiquitous access and context-adapted consumption.

1.3 Network solutions

Network and Content providers have worked together to find an appropriate answer to network congestion. Transmission techniques have been proposed. Those techniques are nowadays widely used to reduce network congestion. Those techniques are a first step to context awareness in networks.

1.3.1 Unicast - Video Streaming on Demand

Unlike the broadcast technique that sends data to every possible destination, the unicast technique sends data to a single known location (Figure 1- 3). Where broadcast is the default technique used in radio or legacy TV broadcasting, it cannot be used in web based delivery since the network will be rapidly overloaded. Unicast is the default technique in web based delivery. In the video delivery context, unicast transmission is used to deliver Video on Demand (VoD) to the end user. It is especially efficient since there are few users that request the same video at the same time. Unicast answers media delivery with a 1 to 1 schema: One data to one end user.

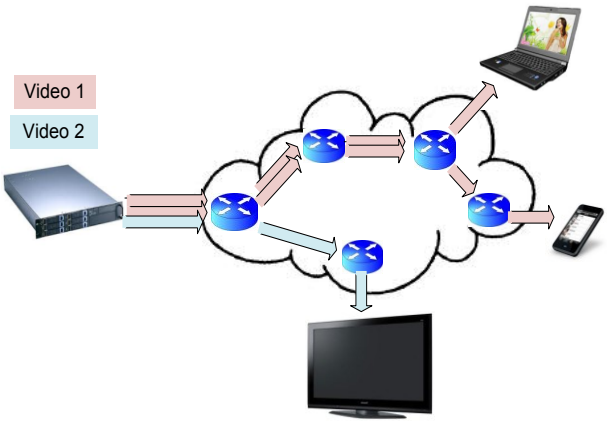


Figure 1- 3 : Unicast distribution Technique

1.3.2 Multicast – IPTV

If "N" end users want to access to the same data at the same time, unicast technique will send "N" times the data, one time per consumer as unicast follow a 1 to 1 delivery schema. Broadcast

technique that answers a 1 to all delivery schemas will overload the global network. Content and Network providers have developed the multicast transmission technique that responds to a 1 to N schema while optimizing network usage. To use multicast, end users have to subscribe to a stream that is send only once. The stream is send to the users and replicated at the last router when the network path between two groups of users split (Figure 1- 4).

Multicast techniques are well suited for live transmission on the Internet, where everyone wants to access the same stream at the same time. IPTV extensively uses multicast.

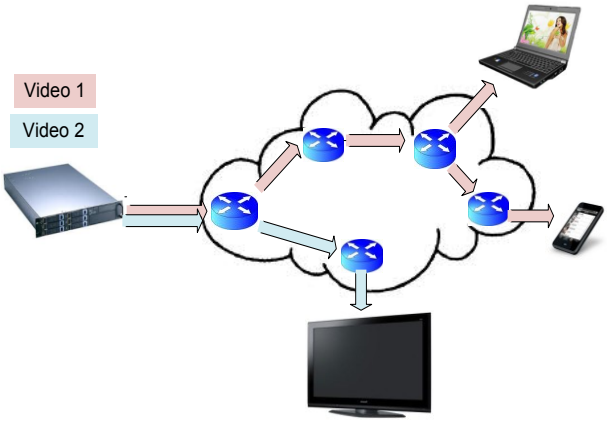


Figure 1- 4 : Multicast distribution technique

1.4 Conclusion

Technology advances have contributed to a true revolution of information distribution. As a result the global network is extensively used and begins to show its limits. The next generation network is seen as being much more content and context aware. Premises of context awareness have been put in action. Efforts have to be pushed further to take into account the heterogeneity of devices used nowadays. The amount of information consumed is exponentially growing due to terminal evolution. The kind of information being consumed is also evolving to be more multimedia based. Content and context awareness have to be extended to this new growing demand.

2 A high multimedia demand

Along with this new type of End-User, another trend is rising: calls are gradually shifting from voice to video-conference; SMS are shifting to more Media Messages, such as MMS but also Facebook/twitter/blogs posts. Therefore, functionalities of mobile devices have been extended to support other mobile usages. Indeed, users want to constantly access Internet services, and most of all they want to deal with multimedia contents. But they do not want to suffer quality loss on their other legacy services (mail, browsing …) because of media consumption.

Considering legacy video consumption, television devices have also been evolving. Connectivity features and multimedia support have been added to TV for a better and more seamless access to his own media - such as one's holiday's pictures or video – or remote media like video on Demand. Television broadcast is now available on computer thanks to IPTV. At home or on the go, video content is more and more consumed.

The most important issue coming from multimedia consumption (video on demand or IPTV) relies in the resources involved for data transmission and data decoding. On the one hand, multimedia video

decoding is a computational intensive task for embedded devices that display it. Moreover, devices that display multimedia are also often handheld devices constrained by their battery life and their processing performances (that are often linked). On the other hand, multimedia data requires high network bandwidth[2]; this can easily overload the network. With more and more multimedia content consumed on the Internet, the network is already showing its limit and proposed solutions are only based on adding nodes and complexity.

Multimedia data and more precisely, **video management and transformation under real-time constraint are a major challenge**. Indeed, video streams are composed of a huge set of information provided in compressed formats. Manipulating and transforming such video streams requires usage of **compression/decompression algorithms** that **are computation intensive** and **difficult to implement for real-time processing**.

2.1 The size of a raw video, a need for compression

A raw video is a video that is not compressed. It is composed of the overall pixel data required to display the complete video at the required frame rate. Nowadays a screen is handling High Definition (HD) video that has a pixel resolution up to 1920 x 1088. Thus, the video has 1088 lines of 1920 pixels each so that it contains:

$$1920*1088 = 2\,088\,960 \text{ pixels} \quad (1.1)$$

In the commonly used RGB color space, each pixel is coded using 3 bytes, one byte for each color component (Red, Green and Blue). Without data compression, a picture weights:

$$2\,088\,960 *3 = 6\,266\,880 \text{ Bytes} \quad (1.2)$$

In order to provide a smooth video perception, a video that is a sequence of individual pictures has at least a display rate of 24 pictures per second. In this case, one second of raw video weights:

$$6\,266\,880 * 24 = 150\,405\,120 \text{ Bytes per second} \quad (1.3)$$

A second of a video takes about 150 mega Bytes. A two-hour raw video weights around 1050 GB, too big for storing on media support (DVD, Blu-Ray), on hard drives or to convey through the networks such as Internet. There is a huge need for video compression techniques to reduce this data amount.

2.2 Multiplicity of techniques

A lot of compression techniques are used today. Techniques can be sorted by types such as transform techniques that change the pixel representation to another one or variable length coding techniques that operate an effective compression of the bit stream. A compression format is defined by a set of technique that together achieves video compression.

2.3 Standardization

Due to the plethora of compression technique that leads to the multiplicity compression format, a need for standardization has aroused and international standards/recommendations have appeared. Two main institutions have gathered experts to create those recommendations. The standardization sector of the International Telecommunications Union ([ITU-T]) came up with a series of

[2] High Definition Video have at least a 25MB/s bitrate

recommendations named h.261, h.262, h.263 and h.264. The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) created the ISO/IEC center that issued the MPEG-X standards [ISO]. The two organizations joined two times to create the renamed h.262/MPEG-2 [MPEG2] and h.264/MPEG-4 standards [H264].

In order to ensure MPEG-X and H.26X to be international standards, the community has to make them supporting a wide range of applications. In order to do so, the standards – or recommendations in the case of H.26X – possess profiles and levels.

A profile describes what features the codec supporting that profile have to be able to decode. While Low profiles are the lightest and generally used for handheld devices that cannot afford to possess a huge and energy consuming decoding chip, High profiles are at the other end of the scope in order to achieve the very best compression rate possible for Television or Movie usages.

Levels characterize the limitation the decoder possesses. On the one end, low level decoders are limited to CIF resolution and low frame rate. On the other end, high level decoders are suited for HD video handling.

A sample of Profile/Level combinations on the MPEG-2/H.262 compression format is given on Table 1- 1 as an example. The low profile on MPEG-2 is named Simple Profile (SP), the middle profile is called Main Profile (MP). Several High Profiles are defined but rarely used since the MPEG-4/H.264 AVC appeared with twice better efficiency. MPEG-2/H.262 is still widely used as the legacy compression format in Digital Terrestrial TV and DVDs.

Profile/Levels	Maximum Resolution	Maximum Frame Rate	Maximum Bitrate(MBps)	Applications
SP@Low Level	176x144	15	0.096	Wireless handheld devices
SP/Main Level	352x288	15	0.384	PDA
MP/Low Level	352x288	30	4	Set-top boxes
MP@Main Level	720x480	30	15	DVD
MP/High Level	1920x1080	60	80	HDTV

Table 1- 1 : MPEG-2/H.262 Profile/Level Combination and their Applications

2.4 Standards and codec

Standards and recommendations are only specifications of how a video has to be coded. What effectively implements the standard is a codec. Codec stands for CODer/DECoder. A codec is a pair of coder/decoder that implements one or several compression techniques at specified profiles and levels. There exists a variety of codec following a unique standard. Figure 1- 5 shows the timeline of existing standards while Figure 1- 6 shows the apparition dates of famous codecs. Most codecs tend to implement H.264/MPEG-4 AVC standard, some purely implement their own compression techniques that are generally related to famous standards.

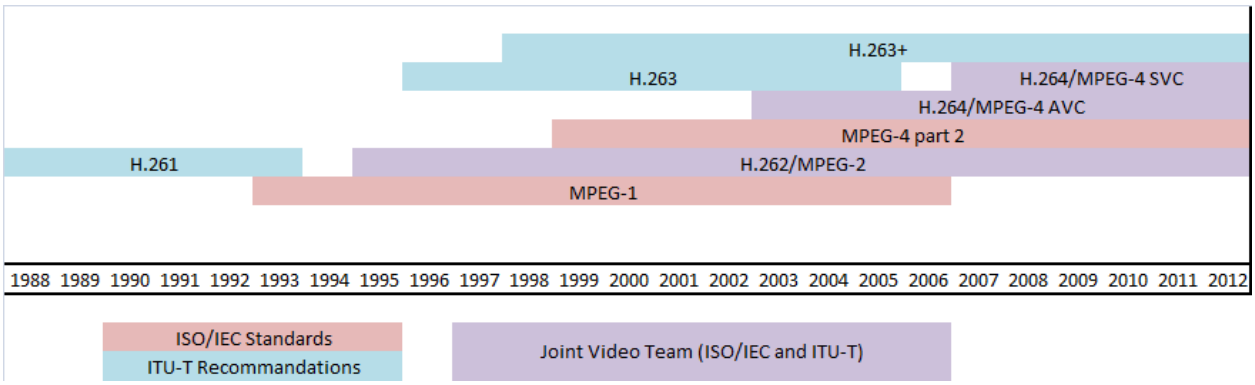


Figure 1- 5 : Standard Timeline

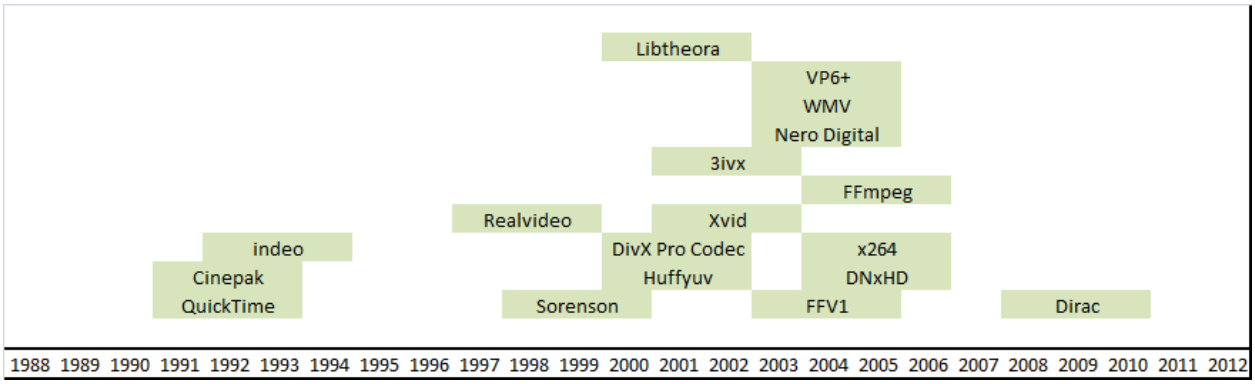


Figure 1- 6 : Codec Timeline

2.5 An always evolving field

Multimedia is an always evolving field. The most recent h.264/MPEG-4 AVC revision has been approved on Mars 2005 and still the complete standard is not been used today. Multimedia actors on the planet are resilient to move to one standard to the other because of the complexity induced by each and every standard. Thus, there is equipment that varies often (smartphone) or is easily updatable (computers) but some are not fit to huge change such as the TV. This is why the MPEG-2 standard is still widely used all-over the world slowly taken over by the h.264/MPEG-4 AVC format.

While industrials are trying to keep pace with new standards, academics keep being prolific on the subject. While doing so, others are already searching for new coding techniques or new way to experience multimedia such as 3D, multi-view [VET11] or High Efficiency Video Coding [HEVC]. This boiling environment that is the multimedia environment confronts the industry to always develop or update their systems; it is not possible to rely on one static structure for it will quickly become obsolete.

2.6 Conclusion

Due to the plethora of coding parameters – codec, resolution, frame rate, bit rate – content providers need to find a way to answer to the various demand. The provider must provide content suited to the end user's context (device, network…) – e.g. iPhone devices only support h264 multimedia contents with their main video player. Providers' answers to this issue fall mainly into three main concepts: video content over-provisioning, multi-layer formats and real-time adaptation.

3 Multimedia content delivery solutions

3.1 Video content over-provisioning

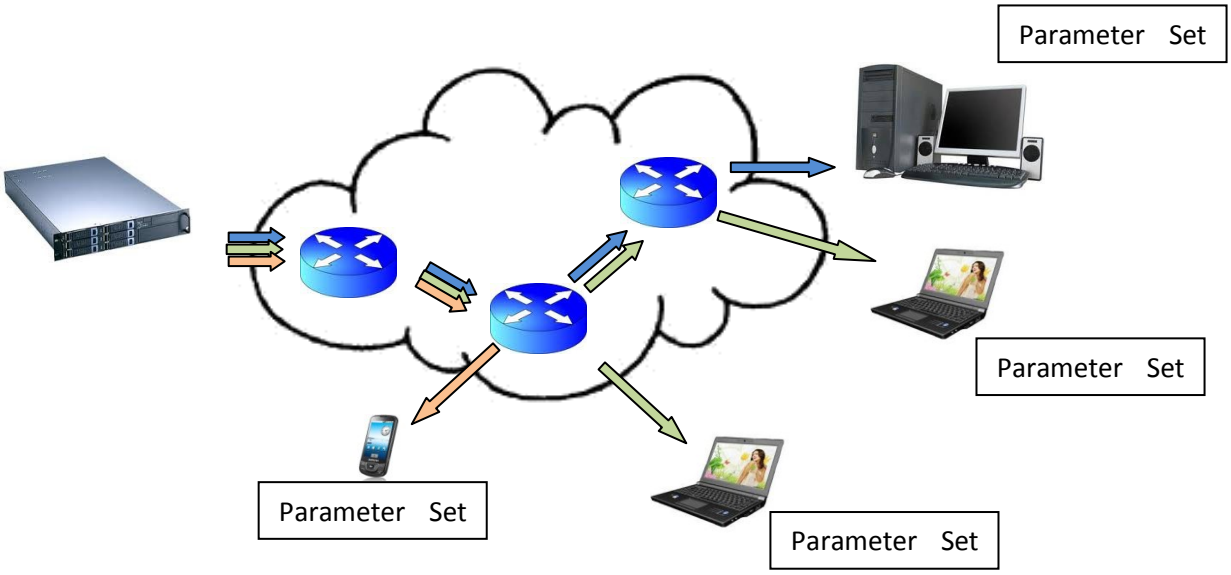
The vision of over-provisioning is a data bank of many versions of one video. Once a customer specifies a set of parameters, the stored version of the video that is the closest to this set is sent. The key of over-provisioning is to size the numbers of versions, along with the different parameter sets, in order to be the closest to the demand.

Over-provisioning is the mostly used technique. It only needs the deployment of data centers and of a few specialized encoders. On the other hand, one video stored requires a lot of memory space due to the important number of versions that have to be stored. Because the Content Provider cannot know which sets of parameters will be requested, he has to create a large set of video versions in order to reduce the distance between what can possibly be asked and what will be provided. Furthermore, the content provider knows only what the request contains, not the device issuing the request. Thus, the video sent is rarely exactly the video required with the proper video parameters set.

On the network provider side, over-provisioning is the action of adding more and more connections in order to provide more bandwidth to the network. It is a short term solution; networks and servers are not infinitely extensible. The ever evolving multimedia field will propose new codecs, new parameters that will need to be added to the provision. Users are not following very fast to a new technical appearance. Thus, over-provisioned server will need a real expansion, not a substitution. Taking as an example the optic fiber installation, the network upgrade is costly and slow.

When facing the broadcasting world, over-provisioning shows its real drawbacks. The multi-cast technique (presented in Section 1.3.2) that aims at optimizing network bandwidth cannot be used at its full potential, because service providers are encoding the multimedia stream with different parameter set to “multi-cast” it to the several network providers (Figure 1- 7) instead of multi-casting one multimedia flow all over the network. This drawback can directly be linked to network congestion as it increase the number of data send on the network and decrease network routing technique efficiency on account of the rise of multimedia stream size and demand.

Some live streaming needs to be encoded in real time. For over-provisioning that means a lot of specialized encoder that process in the same time instead of only one unique encoder.



3.2 Layers and filtering

In order to overcome the multiplication of video file at the server side, layer encoding was proposed [SCH07]. Layer encoding proposes to merge every version of a file into a single-layered file. A base layer encodes the core data of the video. Every other layer contains information that enhances the base layer. Enhancement information consists in spatial information (raising the spatial resolution of frame), temporal information (raising the number of frame per second) or quality layer (raising the quality of each frame in the video). Figure 1- 8 shows an example of different kinds of layer composition, where the original video is composed of each enhancement layer in addition to the base layer.

Upon request, the server can select a subset of enhancement layers to send with the base layer in order to match the request. Another solution is to send every layer and to let the user filter the desired layers. The latter solution means that a unique file is send and thus enable multicast optimization (see section 1.3.2), while the former solution optimize the size of the send stream in a unicast solution (see section 1.3.1).

This vision has been thought with the Scalable Video Coding (SVC) extension of the H264/MPEG-4

Figure 1- 7 : Multicast and over-provisioning

AVC standard. However, H262/MPEG-2 standard has already proposed scalable coding but this feature was not exploited. Today, H264/MPEG-4 SVC is not used by proffesional systems due to the soon upcoming of the next generation standard with better compression ratio. The main drawback of scalable video coding is the time to market issue. Indeed, one has yet to master the existing video standard before beginning to develop the SVC feature of the standard. By the time it is done, another standard, with higher performance, has appeared. To example of this drawback have already been shown. Hence, H262/MPEG-2 scalable feature has not been used for H264/MPEG-4 AVC has appeared and le scalable feature of H264/MPEG-4 AVC is not depoyed due to the up coming of High Efficiency Video Coding arrival.

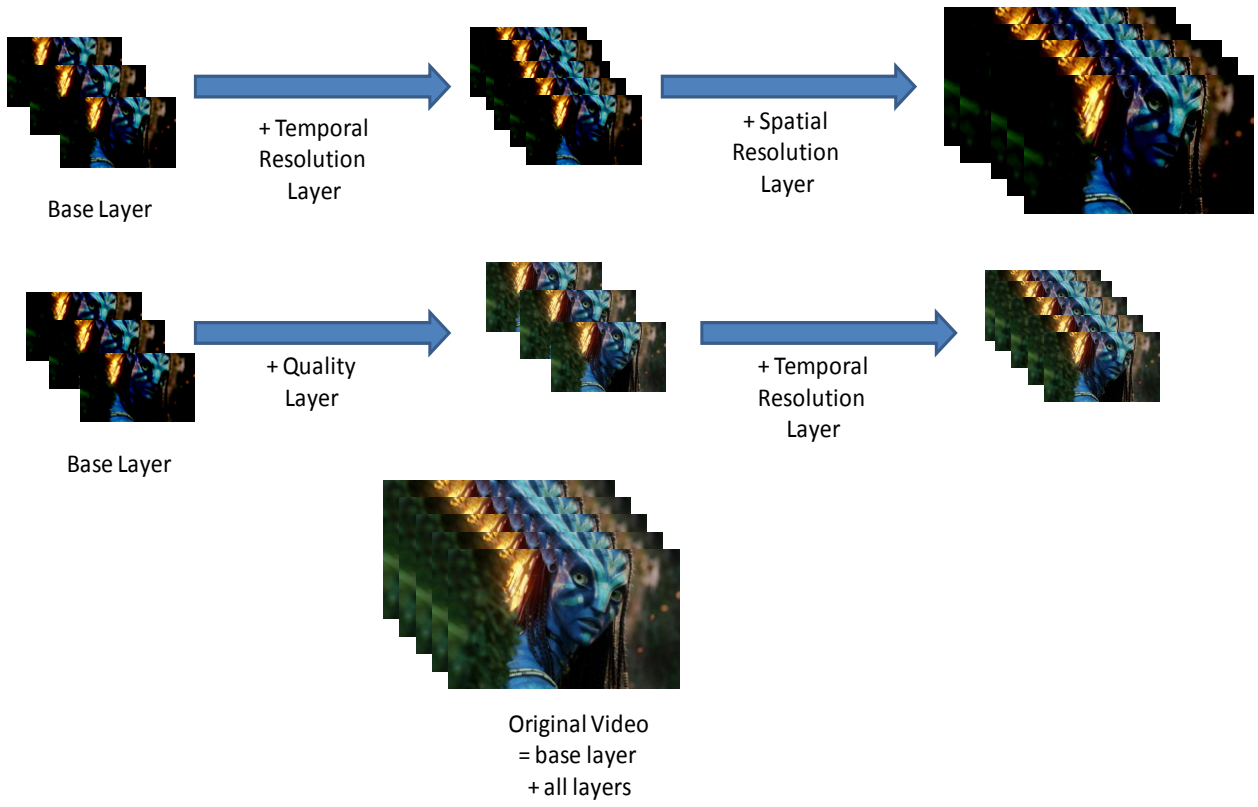


Figure 1- 8 : Layer composition examples

3.3 Adaptation

The adaptation concept is on the other side of the problem. This technique envisions one unique video flow that is adapted throughout the delivery chain (like layer encoding but the video is adapted instead of filtered). Adaptation helps in enabling efficiently the “prosumer” concept foreseen to be of massive usage in Future Internet. The “prosumer” does not have the means to over-provision its own data in order to make it available – the most efficiently - to the community. He must rely on distribution platforms, such as Youtube, Facebook … leaving this way the ownership and legal property of the content, of its usage and management. Then, within the platforms, over-provisioning may be performed. In order to give to the user the possibility to make his content directly available to other users (without losing its rights), adaptation is very beneficial. As well, it may be used by distribution platforms to gain on storage and management efficiency.

Adaptation reduces the stress on over-provisioning by reducing the number of the required versions to one stream to be stored. Thus, less over-provisioned server and a lesser need for increasing network bandwidth may be achieved. One of the most important features made possible by the adaptation is to effectively refine the stream to send in order to match it better with the user context (terminal, preferences, surrounding network, crossing networks quality …).

Nowadays, adaptation techniques can be deployed on different locations of the end to end delivery path. The Service/Content Providers end point, within their servers, can possess adaptation means as well as Network Providers within their routers. There is one place that arouse in the recent years that can also achieve an adaptation process is the last element of the network chain and first element at the End-User premises: the home gateway. This home gateway is, excluding the end user’s terminal, the equipment that can access the fastest and easiest the terminal profile. Thus, the home gateway can decide and operate video adaptation according to context.

Figure 1- 9 shows that having an adaptation at the edge of the delivery path keeps multi cast optimization. While enabling efficient satisfaction of the End-User as a consumer is also satisfies the End-User as a producer in allowing the End-User to keep its ownership. From the Network Provider point of view, network is used in an optimized way (multi-cast, video adapted to the network state) and the Service/Content Provider is no more forced to use additional server to over-provision its content.

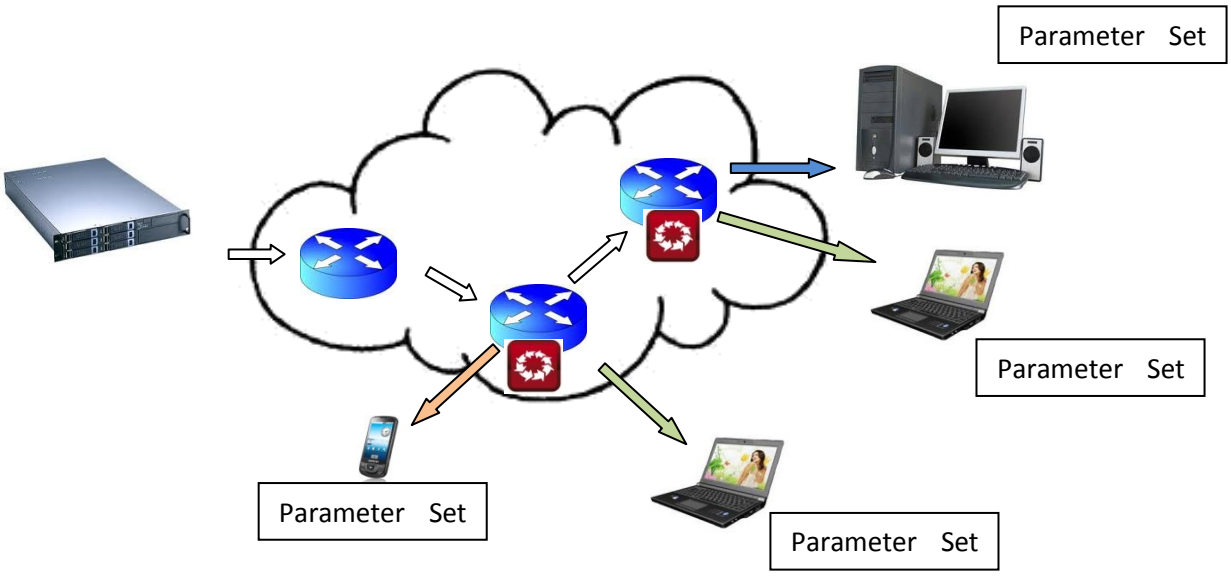


Figure 1- 9 : Multicast and adaptation

Finally, the video adaptation is a well-chosen technique to spread the use of layered video coding. With adaptation, scalable video coding can seamlessly be used inside the network, the video is adapted to a supported codec at the end of the video delivery so the End-User's device does not have to possess SVC decoder.

3.4 Conclusion

While nowadays over-provisioning has been used to deal with multimedia content. Due to the growing demand in this field, over-provisioning shows its limit. Additional content server and network cable deployment is very expensive and cannot follow the rapid growth of data consumption. In next generation networks, adaptation is foreseen as a key feature. This technique is highly scalable, brings better user experience and optimizes network and server usage. The adaptation concept can upgrade already deployed network resources without impairing a technology revolution and hence re-deployment costs. Video adaptation can seamlessly lead to future technology evolution such as SVC or HEVC apparitions without impairing major replacement costs.

While data over-provisioning may still be sustained for Content Provider due to the cost of obtaining and maintaining several data centers, adaptation feature can be deployed at every level of the delivery chain. Thus, it is important to know where the adaptation platform will be optimally placed in order to find the constraint over which the design is created.

4 Towards an adaptation Platform

The multimedia delivery chain (Figure 1- 10) is composed of three main actors: the Content/Service Provider, the Network Provider and the End User. Stream adaptation features must be added at least to one of these contributors.

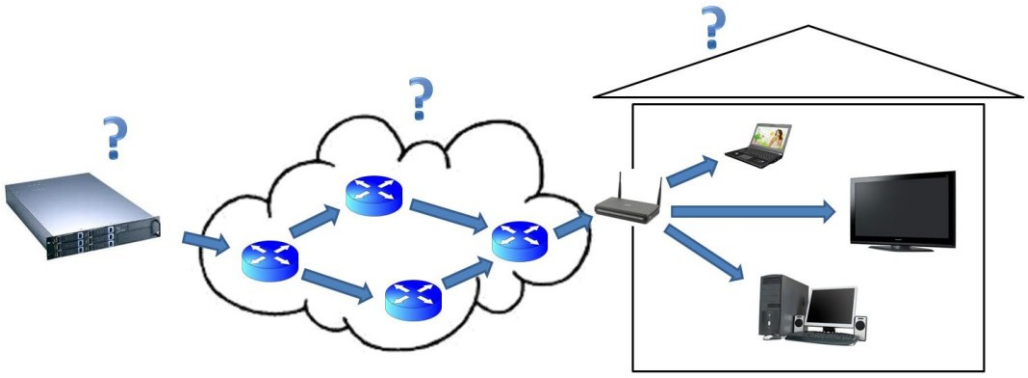


Figure 1- 10 : Possible Adaptation Location

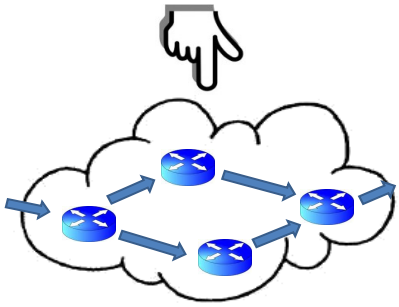
4.1 Where to adapt?

4.1.1 Content/service Provider Device

On the Content/Service Provider side, the adaptation will replace the data over-provisioning technology to deliver the best suited content to the end user. Network will be better handled since only the right amount of data will be sent for each request. Nevertheless, multicast technology will keep on being underused or totally useless for a video will match only one end user. E.g. if every single user requires the same video but with their own particular set of video constraints different from one another, then there will be as many adaptation of the same video and as many sent streams as there are users instead of only one. To keep this scenario from happening, only bitrate adaptation should be allowed which will limit adaptation to the network state.

Furthermore, for the End-User to share his own multimedia content, this solution will require the end user to upload it to a remote multimedia server that possesses such adaptation feature. Hence, the End-User does not keep his ownership.

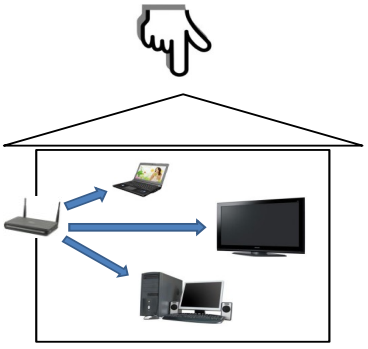
4.1.2 In the middle Network Device



On the network side, adaptation is hard to achieve since all the video packets passes through only a few key routers responsible for the domain switching. But network provider possesses network monitoring tools that enable them to adapt video to the network bandwidth. Putting the adaptation on the network will allow the end user to ask for any kind of video and the content/service provider will provide the video without knowing to whom. In this configuration, the prosumer feature is respected since the end user can put his media on a server of his own, the media will be available to any one since it will be adapted through transportation process. This choice put a lot of stress on the network provider which not only has to maintain a lot of processing resources on key routers but also has to deploy a huge intelligence network machine to take decision on where the stream has to be adapted. The problem tackled by putting adaptation inside a network router is twofold:

- network nodes, as they are today, are not capable of knowing what type of flows are passing through them. They only know that it is an IP packet and its destination address. A complete change in the IP routing paradigm is necessary in order to enable an adaptation process inside routers.
-
- adaptation is an heavy task that requires much more resources than what is being done by a router today, which is supposed to only deal with routing and forwarding functionalities (especially in the core domains).

4.1.3 End User Device



The last but not the least option is to put the adaptation feature on the End User equipment. On the center of the End User's network at the near end of the multimedia flow lies the home gateway. This equipment is responsible for creating the network inside the home and to connect this network to the outside network in order for the customer to receive the services he has paid for. The prosumer concept has already arisen with the remote server capability given to the home gateway. When the user wants to make his data remotely available, this feature allows keeping one's own data home instead of giving it to a content provider. In the network provider benefits from fully optimized multicast techniques since the stream will only change in the end of the delivery chain.

Where evaluating the adaptation process load at the server side, the number of End-User per gateway is limited for there is one gateway per home and a limited amount of users inside the home (i.e. the family members). Hence, having adaptation capabilities at the home gateway side is very scalable as there always are enough home gateways by potential users.

4.2 Home Gateway constraints

Today, business models in xDSL/Fiber domain for providing Internet connectivity to the End-User implies extensive use of home gateways as the last hop router in the content delivery over network. Home gateways are provided by network provider to end user. This equipment allows the connection to the network provider domain while creating a home network. The huge number of devices lent to the customer and the fact that it is the network provider property is the reason why network providers try to keep its cost low.

That is why the adaptation platform in the home gateway has to be kept low-cost. If this constraint is achieved then this platform could as well be put on access networks equipments (such as 3G Base Stations or WiFi Access Points). Moreover they can even substitute to server required for data over-provisioning as they will adapt the content that was previously duplicated by content/service providers.

The adaptation process shall not limit the streaming process. Indeed, if the adaptation process does not follow real-time constraints, lagging issues will appear. Furthermore, every family member should benefit from this feature and thus the adaptation process should be able to schedule multi stream delivery.

5 Software limitations

Video adaptation is an intensive task and requires a lot of processing power. Therefore, if video adaptation is performed by software programs, it will have to run on powerful processors to achieve real time constraints. However, powerful processors are not targeted in the embedded industry because of their high costs. Embedded systems are generally composed of a low cost processor that performs basic tasks or micro-controller that drives hardware accelerated process. Hardware computing is 10 to 100 times more efficient than software computation. Hence, we envision video adaptation features as hardware accelerated process.

6 Hardware possibilities

Hardware designs target two kinds of chips [RAJ01]. First and foremost, the Application Specific Integrated Chip (ASIC) is the main target of hardware development. ASICs are integrated circuit specifically designed to perform a specific task. Once the ASIC component is developed, it is locked and no evolution to further tasks is possible within the component. However, this static drawback comes with a great execution speed and efficiency in terms of cost. It is targeted to a wide market as the fabrication process has a high initial fabrication cost (~500 000$ to 1,5 M$ for creating a 90nm mask) and a very low fabrication cost per unit (~10$ to 30$ per unit).

Secondly, the Field Programmable Gate Array (FPGA) is a configurable hardware chip. This configuration feature allows changing the design circuit that the FPGA operates. Thanks to this flexibility, the FPGA chip is mainly used by ASIC designers for reducing the prototyping cost. FPGAs have lower performance than ASIC and targets small market as each chip has a fixed cost (from 30$ to 5k$ per unit).

7 ARDMAHN Project

The same motivations have led to the creation of the ARDMAHN project. The Dynamically Reconfigurable Architecture and Methodology for Self-Adaptation in Home Networking project (in french: **A**rchitecture **R**econfigurable **D**ynamiquement et **M**éthodologie pour l'**A**uto-adaptation en **H**ome **N**etworking) is funded by the French national research agency (ANR) in the 2009 ARPEGE program (project id: ANR-09-SEGI-001). The ARDMAHN project aims at solving this adaptation issue by using the partial dynamic reconfiguration of the new generation of FPGA. Dynamic reconfiguration is seen as a tool to reduce FPGA cost. Dynamic reconfiguration enables to reconfigure parts of the FPGA chip at runtime. Thus, hardware tasks can be configured in the FPGA only when needed, optimizing the FPGA resource space. While video adaptation task remains the duty of hardware accelerator, networking capabilities (packet managing …) remains the processor duty. Interfacing processor with hardware accelerator reduces the hardware overall efficiency mainly due to data transfer rates and synchronization issues. Also hardware dynamic reconfiguration is still at its very beginning and is not yet well mastered. Hence, the ARDMAHN project aims at (1) studying the feasibility of embedding video adaptation process in a low-cost home gateway while (2) finding a method to design process on a dynamically reconfigurable hardware.

8 Conclusion

Today's Internet network is flooded with media content coming from professional Content Providers as well as Prosumers. In order to keep up with this high demand, the network has to evolve from a content agnostic type to a more content and context aware type. By doing so, the network will be able to cope with the emerging "prosumer" and consumer on the go by managing network adaptation considering both content and context. In this paradigm, video content is one of the more network impacting content because it is one of the most consumed and the most bandwidth requiring.

To answer these limitations, a next generation network is envisioned: the Future Internet. This new network is content – and context - aware. To ensure this awareness and enable new services and capabilities for all the actors (SP/CP, NP, End-Users), especially for video content, video adaptation is foreseen as a key solution. Video adaptation platform shall mainly be embedded in low constraint network devices such as home gateways.

In order not to limit the streaming process, the adaptation must follow real-time constraints. However, video manipulation is a computational intensive task that operates on huge amounts of data. Hence, video adaptation cannot be operated on processor as it will require expensive powerful processor. As a result, real time and low cost constraints can be met by hardware accelerated technology. These considerations motivate our work. In the next chapter, an overview of state of the art video adaptation systems are presented.

Chapter 2 : State of the art on video adaptation

1 Introduction

The Internet is evolving towards a multimedia centric ecosystem. In this new ecosystem, video adaptation is foreseen as a key feature to content and context awareness. The location of such adaptation has been envisioned in the last hop routers – home gateways, 3G base station … - that are designed with low cost constraints. Therefore, designing a low cost video adaptation process is mandatory. This second chapter is decomposed in two parts. First and foremost, video compression basis are explained. Thanks to this knowledge, in the second part, a video adaptation system overview is then presented.

2 Picture and video compression techniques

2.1 Compression techniques

In most devices, picture information is captured or processed using a RGB representation. In this representation, each pixel of the picture (for a “true color” picture) is encoded using three distinct values that represent the levels of red, green and blue information. An example of such RGB format is provided in Figure C. Each value is often stored on 8 bits, so a single pixel is coded using 24 bits.

A nowadays camera taking 15 mega-pixel pictures (pixel dimension of the picture is 4752 x 3168). The amount of memory required to store a picture is 4752 x 3168 x 24 bits. RAW data are impossible to store as they are, so compression techniques have been developed for picture storage. These techniques can be lossy or lossless.

Video content tackles the same storage issue, even if (HD) video frame resolutions are lower than 15 mega-pixel pictures, at least 24 frames per seconds are required. For example, a High Definition (HD) video has a resolution of 1920 x 1088 pixels at 30 frames per second. A frame contains 2 088 960 pixels. In true color RGB format, a single video frame weights 6 266 880 bytes. To store only one second of an HD video more than 150 mega bytes are required. Hence, compression is needed to store and transmit video.

Many compression techniques and standards have been proposed to solve picture and video storage or transmission. These compression processes are based on redundancy elimination through mathematical transformations. The decompression process aims at retrieving the eliminated information through inverse transformations. The following sections quickly present the transformations used by picture and video compression techniques.

2.2 Picture compression – The JPEG standard

The main objective of picture compression techniques is to reduce as possible the amount of information required to store and to delier the picture while minimizing quality loss. Regarding this objective, compression techniques have been developed. The JPEG compression is one of the most used and best known. This technique which principles are reused in video compression standards such as MPEG-x or h26x is composed of multiple data transformations as shown in Figure 2- 1.

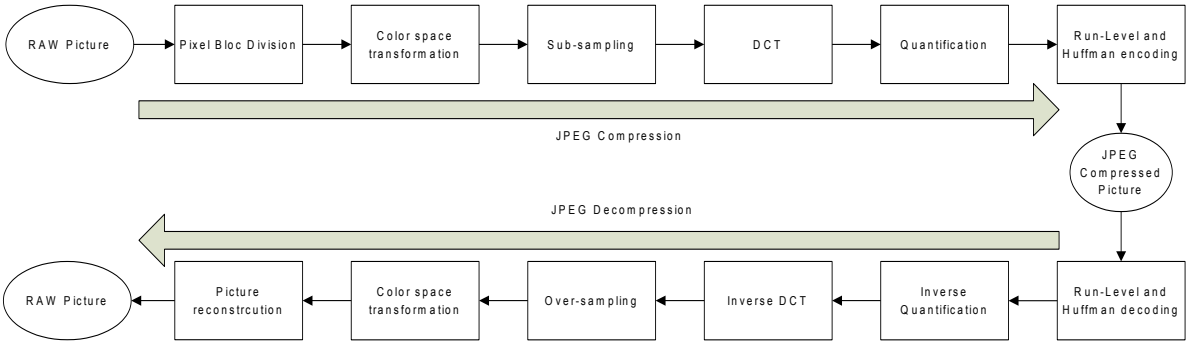


Figure 2- 1 : JPEG Compression Flow

The RGB color space is inefficient for picture and video compression. Indeed, channel information are correlated. So the first lossless transformation comes from the picture color space. RGB color space is converted to another color space : the YCbCr one [BT601]. This color space uses as brightness (Y) and the chrominance (Cr, Cb) decomposition of the colors. An example of color space transformation is provided in Figure 2- 2.

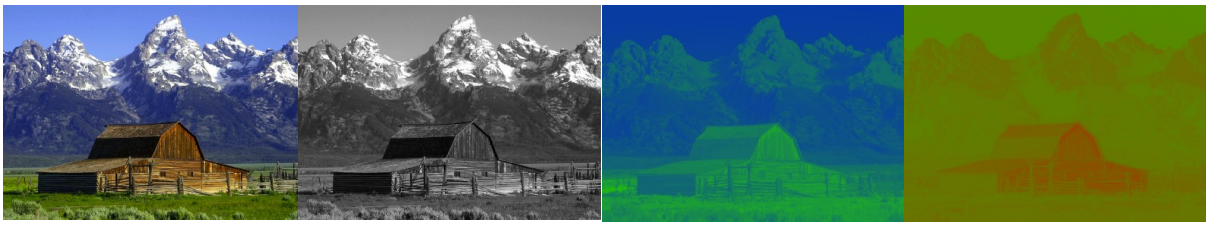


Figure 2- 2 : (a) Orignal RBG Picture, (b) Y channel , (c) Cb channel, (d) Cr channel

The advantage of using the YCbCr is the use of psycho-visual behaviors that make the human more sensitive to luminance (Y) factor than the chrominance blue (Cb) and red (Cr). An example of such assertion is provided in Figure 2- 3, where (b) and (c) have been transformed in an indentical way. However, in Figure 2- 3 (b), Cb and Cr channels have been blurred, when in Figure 2- 3 (c) only the Y channel has been modified.

Using psycho-visual fact, the overall Cb and Cr information are subsample by a factor 2. In this case, the YCbCr format is named 4:2:0 (without subsampling it is named 4:4:4). In a 4:2:0 representation, the number of information that must be treated has been halved compared to the 4:4:4 representation, as shown in Figure 2- 4.

Figure 2- 3 : (a) Original RGB picture (b) Picture with a Cb and Cr blurring (c) Picture with Y blurring

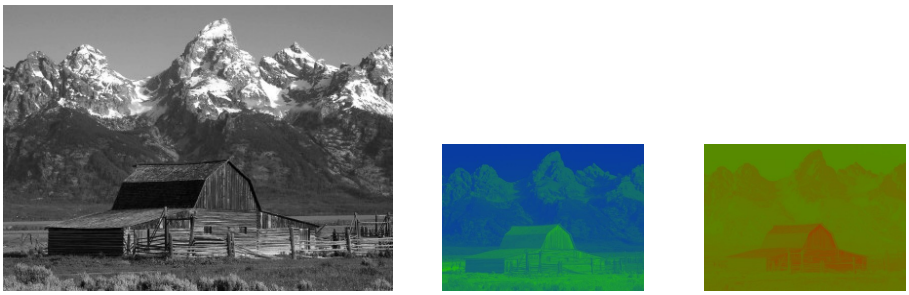


Figure 2- 4 : Picture in 4:2:0 mode (a) Y channel (b) Cb subsample channel (c) Cr subsample channel

For the rest of the JPEG compression flow, the complete picture is not processed in a single step: Y, Cr and Cb channels are processed independently (with the same algorithms). Moreover, the picture is splitted in blocks of 8x8 pixels (macroblock) to reduce the computationnal complexity. In a 4:2:0 representation, 4 blocks of luminance (Y), 1 block of red chrominance (Cr) and 1 block of bluechrominance (Cb) are grouped to form a macroblock (MB).

Spatial pixel representation is not the most efficient one for picture compression. Each macroblock composing the picture is converted in the frequency domain using a two dimension discret cosinus transform (2d-DCT). Data in the resulting MB represent the frequency coefficients of the original MB. The first coefficient (top-left) is the continuous (lowest frequency) coefficient called the DC coefficient and the highest frequency one is the last one (bottom right). Figure 2- 5 shows a Y macrobloc in the spatial domaine (a), where data correspond to the Y component values and the corresponding frequency values (b).

To reduce the frequency information amount to be stored, a quantization step is applied. This quantization step divides each DCT coefficient with a value that increases according to the coefficient frequency. Indeed, high frequencies represent small details and low frequencies represent the coares structure of the picture. Hence, high frequency damages are less noticed by the human eye than low frequency damages. The quantification transformation induces information loss that less impact picture quality. Depending on the quantization matrix, the compression ratio and the picture quality vary. Figure 2- 6 shows an example of a quantification matrix (a) and the resulting Y block after the quantization step (b).

The quantization step helps in reducing the amount of information to be stored. Indeed, dividing coefficients by lowers its value. When the value is too close to zero it is considered as zero. The quantization process increases the number of zero coefficients that are efficiently compressed in the

$$f = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix} F = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix}$$

Figure 2- 5 : DCT operation (a) original spatial Y block (b) resulting frequency Y block

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} F^* = \begin{bmatrix} 79 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2- 6 : Quantization operation (a) quantification coefficients (b) resulting Y block

following steps. Zero coefficients are easily compressed by the next and final compression steps.

Finally, the information set is reordered using a zigzag scan order to align the maximum number of zero values. Different zigzag scan orders exist. The most used zigzag scan order is shown in Figure 2-7.

Finally, a variable length encoder is used to factorize the reordered data set. For the current example, the compressed macrobloc stream is *{79, 0, -2, -1, -1, -1, 0, 0, -1, EOB}* where EOB notices that the last elements of the MB are zero values. Static Huffman table (arithmetic encoding) is then applied on the overall picture variable length encoded information.

$$F^* = \begin{bmatrix} 79 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

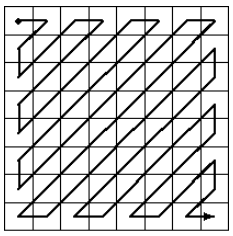


Figure 2- 7 : Zigzag Order (a) Y block (b) Zigzag order

2.3 Motion Picture Coding – The principles

A video is a serie of pictures that are displayed at high speed to give a continuous motion impression due to retinal persistence. Hence, the video compression process uses features that are unique to motion picture such as time resemblances: pictures are often closed to the next ones. An example of two consecutive pictures in a video sequence is provided in Figure 2- 8.

Figure 2- 8 : (a) Reference picture t (b) Picture to be compressed t+1 (c) difference between pictures

Compression algorithms use techniques to avoid the complete encoding of two similar pictures: the second picture is described as the previous one plus some modifications. This approach helps in reducing efficiently the number of information used to encode a video stream. The differences between two pictures are mainly due to motion: object motion (e.g. a moving car), object deformation (e.g. a grappling hand) and camera movement (e.g. traveling). Those motions must be estimated. Motion estimation reduces the number of residual information which implies less information to be coded and thus a better compression is achieved.

Two types of information are used to describe the second picture relatively to the reference one: the motion vectors and the residual information. The motions vectors indicate pixel block motions between the reference picture and the to-be-encoded one. Differences between two pictures cannot only be expressed by motion translation as object can: (1) move further away from the camera or (2) be revealed by a shifting object in the foreground. As a result, motion information shall be completed by residual information to form the new picture. The residual information correspond to the pixel value errors between the to-be-encoded picture and the motion compensated reference picture.

Video standards define 3 types of pictures in a video sequence:

- Intra pictures (I pictures) are reference picture that do not use any reference (motion vectors) to be compressed. They are used for first picture in the video stream and at key point to allow fast forward feature. Because they do not require references, they are compressed using only picture compression technique. They provide the best quality in the stream but they have a low compression ratio. They are used as reference for the two other types of pictures;
- Predictive pictures (P pictures) are compressed using reference (I or other P pictures). They contain motion vectors and residual information and provides a higher compression rate but have a lower quality than I pictures. They can be used as reference for other P or B pictures.
- Bidirectional pictures (B pictures) are compressed using two references (I or P pictures). They use up to two previous or following reference pictures unlike P pictures that only use a unique reference. B pictures provide the highest compression ratio but the lowest quality of the three types. They are never used as reference.

A video sequence is encoded using these three types of pictures as shown in Figure 2- 9. The number of I, P, B frames varies from one video to another dependeing on various parameters such as the video content, the targeted bitrate … e.g. for video with fast motions, more I frames are used to maintain a high video quality.

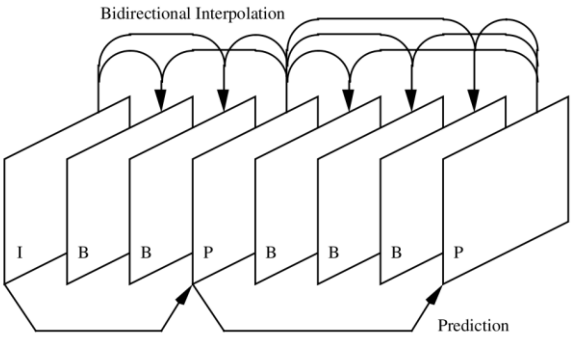


Figure 2- 9 : Group of Pictures [MPEG2]

The main issue with the reference/prediction technique is the error propagation also known as error drift. If an error occurs in a picture that serves as reference, it will be propagated to the next picture. This picture may also be used as a reference for another one. The error will drift. In order to avoid such propagation, I pictures are regularly coded as they do not use any reference picture and thus will stop the error drift. The picture series coded between two I pictures is called a Group Of Pictures (GOP).

2.3.1 Motion estimation in the video encoder system

The quality of the motion vector impacts the bitrate/quality encoding video tradeoff. If a motion vector is not well chosen, it will miss the real macroblock motion and residual information will be generated. More residual information leads to a lower quality due to the quantization step that will lose some of these information or higher bitrate (if the encoder has to keep a high quality).

However, evaluating the motion of every pixel in the picture is a very high computational task for the video encoder system. It is responsible of up to 50% of the computation required to compress a video sequence. Indeed, motion vectors shall be found for each macroblocks that represent a 16x16 pixel region of the picture. To reduce the computation complexity, motion vector is searched over a reduced window instead of the whole picture. Moreover, to reduce the computation complexity, in most case, heuristic algorithms are used to execute only a subset of the possible motion estimation computations. Some of the proposed solutions are the cross [CHA07] and diamond search [LIU07] algorithms that are iterative algorithms. The main idea is to reduce the computational cost in finding not the best motion vector - that minimizes the residual inforamtion - but only a good motion vector.

In order to maintain a high video quality, nowadays video codecs incorporate more sophisticated techniques. Indeed, we said previously that the inter pictures (P or B pictures) are encoded only using motion vectors. However, this approach is not efficient when smaller parts of the picture have high-speed motions. To solve this issue, picture type have been refined to macroblock type. For example, P pictures can be composed of either inter or intra coded macroblocks depending on the quality-compression rate tradeoff. Figure 2- 10, Figure 2- 11 and Figure 2- 12 illustrate the aforementionned techniques by providing different view of two consecutive pictures from a video sequence. The first one is a I picture that serves as reference for the next P picture.

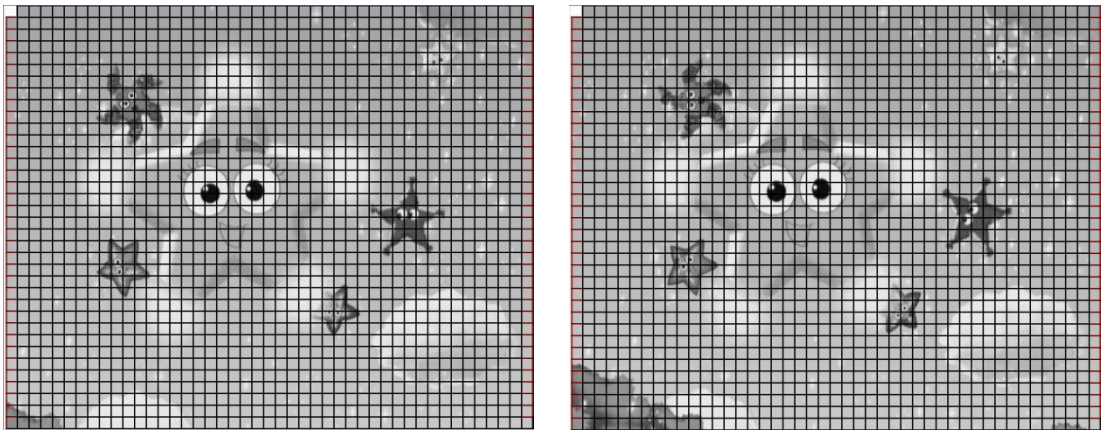


Figure 2- 10 : Two consecutive picture from a video stream

Figure 2- 10 shows the two pictures cut in small 16x16 luminance blocks. The second picture that is P encoded is not composed only of inter coded macroblocks. As shown in Figure 2- 11, the type of encoding selected by the video encoder for each picture sub-bloc may vary according to its compression efficiency and the selected bitrate/quality tradeoff. Encoder choices depend on the motion vector found, the number of residual information still to encode and the distortion induced by each choice.

Depending on the coding decision (macroblock type, motion vectors, …), the amount of residual information to compress varies. Figure 2- 12 shows the amount of residual information required to be computed for macroblocks in the picture (darker blocs require the lower amount of information).

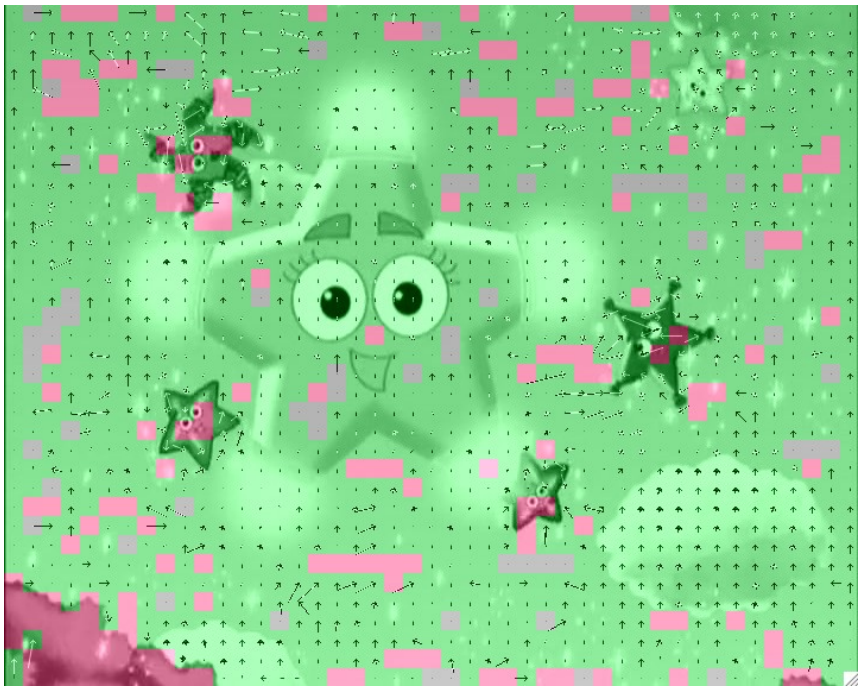


Figure 2- 11 : Second picture encoded (Green MB show Backward Predicted MB, light pink show Forward Predicted MB, darker pink MB show Intra coded MB and arrows show motion vectors).

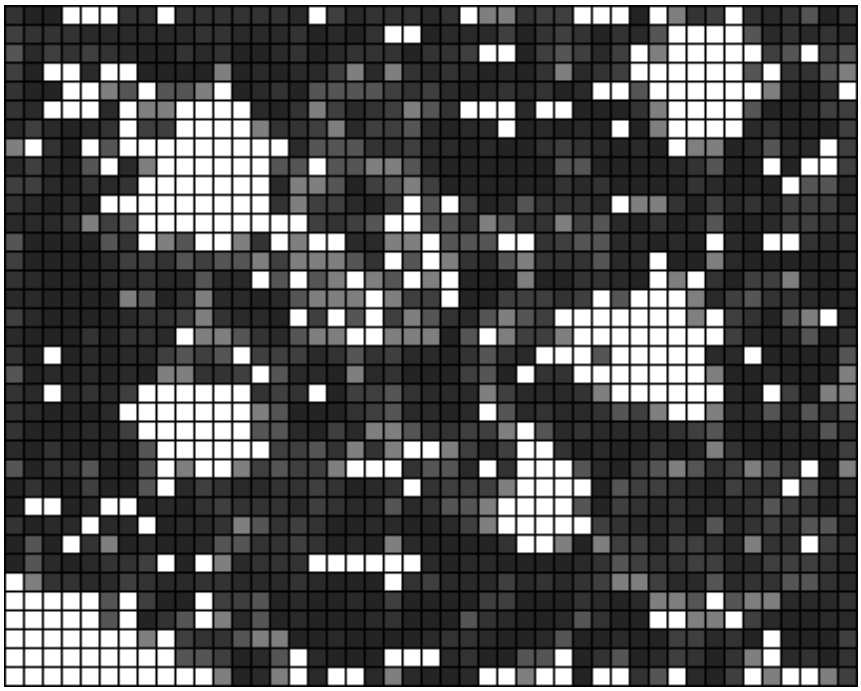


Figure 2- 12 : Amount of information required to store picture macroblocs in the bitstream (white color means costly MB and black color indicates low-cost MB).

2.3.2 Video stream format for data streaming

However, modifying a compressed video stream in realtime is a complex task. Indeed, video information is not directly available. Video streaming standards propose data stream encapsulation in order to mix video, audio and control information. Figure 2- 13 shows the different structures, from the uncompressed video stream, captured by the camera, to final transport stream sent to the end user. Every step adds its header to help synchronizing or decoding the rest of the packet. Thus, some stream information are buried deeply in the successive layer. Segmentation and packetization can lead to split pictures in different packets. Accessing information on the fly can be extremly difficult without a decapsulation (for general information) to a full decompression (for pixel information). i.e. for picture dimension adaptation. The spatial dimension information is contained inside the video header but to process a resizing, the pixel information is mandatory. To access pixels information, the decapsulated data must be fully decompressed.

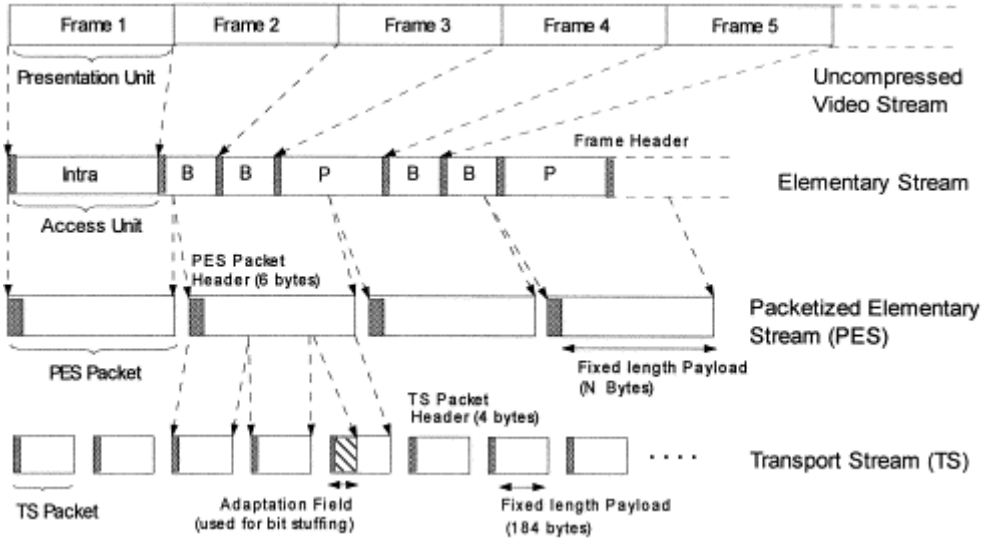


Figure 2- 13 : Data structure in a MPEG-2 stream

The aformentionned compression techniques that have been presented are used by every standard (ISO/IEC or ITU-T) but parameters vary from one another. e.g. H.264 uses ¼ pixel accuracy for its motion vector where MPEG-2 only uses ½ pixel accuracy. Other techniques may be used such as intra prediction technique for H.264 that will not be discussed here.

3 The different types of video adaptation

The video adaptation process aims to transform the video stream in order to change the video data characteristic to match a desired set of constraints. Compressed video streams are characterized by four main parameters:

- The *spatial resolution* – it specifies the number of rows and columns of pixels that compose the video pictures. The spatial resolution of a Full HD video stream is 1920 x 1080 pixels (columns x rows);
- The *temporal resolution* –also named frame rate, it indicates the number of frames (i.e. pictures) that must be displayed per second by the video decoder. There are three main frame rates used for video and TV diffusion: 24fps (frames per second), 25fps, and 30fps;
- The *codec* – The video codec characteristic (**co**der-**dec**oder) indicates the specific implementation of the techniques presented above that have been used to encode the video. Hence, it also specifies decoding techniques to use to decode the video. It also specifies the associated profile/level that are required to decompress the video into a displayable picture sequence (see chapter 1). Digital Terestrial Television (DTTV) broadcast TV programs using H.262 and H.264 video compression standards;
- The bitrate – it indicates the number of bits per second that are used to encode the picture sequence. Bitrate is dependent on the spatial resolution, frame rate and codec used but if those three parameters are fixed, bitrate indicates the overall quality of a video (for the same video sequence). e.g. bitrates lower than a hundred kilo bits are used for low-quality or low spatial video where bitrate for professional applications can be up to hundred mega bits per second. Video bitrate is generally constrained by the network that is used. Full HD video stream transported by the TNT[3] have a bitrate from 1 to 10 some Mb/s.

[3] http://www.digitalbitrate.com/

Video adaptation is a process triggered to answer specific needs. Each type of adaptation is used to answer a typical need such as:

- Network congestion is answered by bitrate/frame rate reduction to fit the available bandwidth;
- Limited terminals capacities (supported codec/spatial resolution) are answered by codec and spatial resolution adaptation.

Another adaptation technique is frame rate adaptation, which is mainly based on frames removing. In order to facilitate the adaptation, B frames are mainly selected for being removed, so that there is little impact on the global stream. Indeed, B frames are never used as references. B frame is also the frame type that achieves the best compression ratio. Removing a B frame will have little impact on the overall bitrate of the video. Furthermore, nowadays video are coded with a frame rate up to 30 frames per second. The human eye notices video discontinuity when the frame rate is lower than 24 frames per second. Thus, reducing the frame rate will be effective at most when reducing this characteristic from 30 fps to 24 fps. This operation removes (30-24)/30=6/30=1/5th of the video frames which frames will be mainly B frames. As result, there are little room of improvement using frame rate adaptation. Hence, frame rate adaptation will not be tackle in our work.

4 Genericity for video adaptation

In order to support the video adaptation[4], the adaptation system must be composed of at least three main processing blocks (Figure 2- 14):

- The video decoding block that decompresses the video stream in codec C_1 into a video raw format;
- The video adaptation block that transforms the video characteristics;
- The video encoding block that recompresses the video from the raw format to the output codec one C_2 at the new expected bitrate.

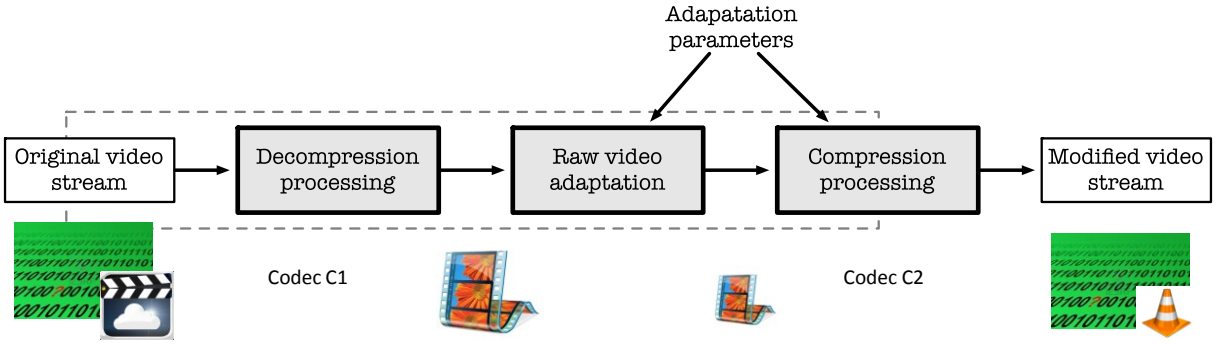


Figure 2- 14 : Video Adaptation Framework

[4] Network decapsulation for streamed video stream is not considered here. This stream processing is realized prior and futhur to the adaptation processing.

This framework derives from the traditional adaptation system used as reference. To our knowledge, there is no adaptation system that does not follow this framework. The traditional way to deal with video adaptation is to fully decompress the data stream that contains the input video, to adapt video characteristics to the specified constraints (in raw format) and to fully re-encode the video in the desired codec.

This straight forward adaptation system is called the *reference adaptation system,* it is used as a reference for video processing algorithm proposal. It (Figure 2- 15) is composed of a complete video decoder cascaded with a complete video encoder.

The system can perform any adaptation combination. The decoder transforms the compressed video into a raw version (completely decompressed video data). The raw stream is easily transformed by the adaptation process that covers temporal (removal or addition of frames) or spatial scaling conversion. At the end point of this processing chain lies a complete video encoder. This complete encoder enables the encoding into the specified codec (same codec or not). As any complete encoder, it is also responsible for controlling the required bitrate, thus covering all four kind of adaptation process, presented above.

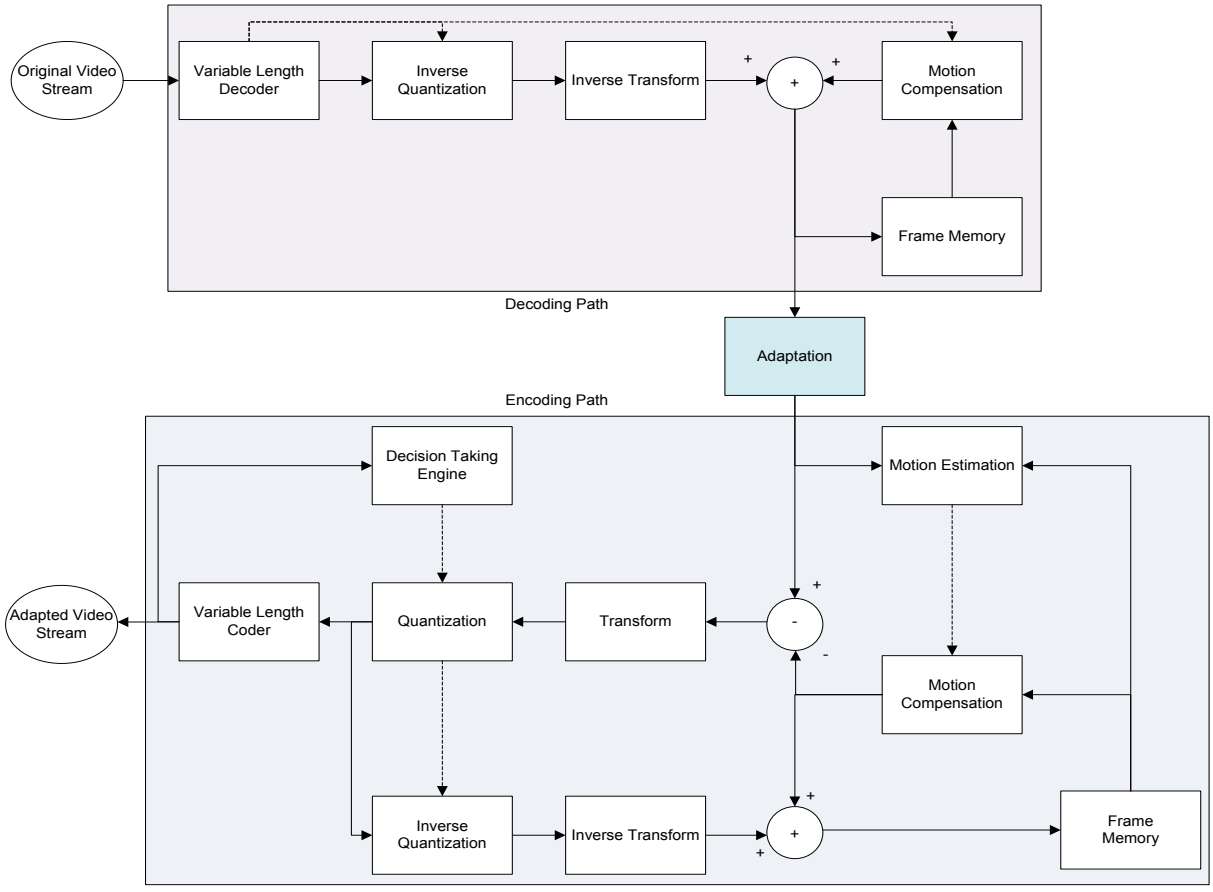


Figure 2- 15 : Reference Adaptation System

The complete encoder is composed, mainly, of the processing blocks required to realize video compression [MPEG2], as well as the motion estimation and macroblock type decision ones. These last two blocks are the most costly ones in the encoder. Their functionality is to assure an optimum compression ratio – i.e. to achieve the best quality while meeting the given bitrate constraint. This is why this adaptation system is used as a reference for other proposals ([AHM05] and [XIN05]). Proposals aim at being less computational while achieving optimization at the closest level as the reference system.

The reference adaptation system (Figure 2- 15) is quality efficient but has a major flaw: full video decoding followed by full video encoding are highly computational. To find the best parameters, decision taking engine and motion estimation blocks require a huge number of computations (e.g. to find the best motion vector motion estimation has to look at every possible motion). Software processing time is directly linked to the computational complexity. Hence, realtime video adapation using this approach cannot be achieved for HD video using nowadays low cost processors as detailed in Table 2- 1. This table has been obtained by performing different video adaptation using FFmpeg on an ATOM Z530, 1.6 MHZ with the Poulsbo chipset. The performance are quantified in the Sw fps column, which indicates the maximum frame per second that can be processed in software. Outlined raws achieve real time constraint (the 10th raw achieves 24 and 25 fps but not 30fps). The only way to reach realtime performances on every use case, is to use dedicated hardware circuits, which cost are area dependent. However, due to the high computation complexity, these circuits may require important silicon area and power consumption.

#	Adaptation	Input			Output			Sw fps
		Résolution	Codec	Param.	Résolution	Codec	Param.	
1	Bitrate	720x576	MPEG2	2Mbps	720x576	MPEG2	1.6Mbps	56.6
2	Bitrate	720x480	H.264	2Mbps	720x480	H.264	1.6Mbps	2.5
3	Codec	720x576	MPEG2	2Mbps	720x576	H.264	1.6Mbps	2.5
4	Codec	720x576	H.264	2Mbps	720x576	MPEG2	1.6Mbps	32.9
5	Resolution	720x576	MPEG2	2Mbps	360x288	MPEG2	0.5Mbps	72.7
6	Resolution	720x576	MPEG2	2Mbps	180x144	MPEG2	0.125Mbps	93.0
7	Resolution	720x576	H.264	2Mbps	360x288	H.264	0.5Mbps	8.0
8	Resolution	1920x1080	H.264	16Mbps	960x540	H.264	4Mbps	0.6
9	Codec + Resolution	720x576	MPEG2	2Mbps	360x288	H.264	0.5Mbps	8.6
10	Codec + Resolution	720x576	MPEG2	2Mbps	180x144	H.264	0.125Mbps	28.0
11	Codec + Resolution	720x576	H.264	2Mbps	360x288	MPEG2	0.5Mbps	39.1
12	Codec+ Resolution	1920x1080	H.264	16Mbps	960x540	MPEG2	4Mbps	5.3

Table 2- 1 : Software Performance using the refertence adaptation system with FFmpeg on an ATOM Z530, at 1.6 MHZ

This limitation has been the motivation for novel adaptation solution, mainly in case of realtime video exploitation. The main idea that leads to new proposals is to reuse information contained in the incoming video stream. Some input stream information are decoded and reused in the encoder path to reduce its complexity (i.e. removing some computational blocks). However, input stream information cannot be reused as they are, low complexity computations are now required to adapt them to the video encoding path. This information reuse operates heuristics and may lead to quality degradation of the outgoing stream due to approximations, the goal in designing these heuristics is to minimize quality loss and complexity computation.

As an example, let's consider a factor 2 video downscaling transformation. This transformation halves by two the spatial resolution of the input video. Each set of 4 macroblocks (2x2) of the input video are merged in order to produce the newest one (Figure 2- 16). To discard computations required to obtain motion vector for the new macroblock during the compression process, one solution is to reuse the motion vectors of the merged macroblocs (ones contained in the input stream). To compute the new motion vector value by using the four existing ones, there exist many solutions (e.g. the new motion vector can be an average of the four old ones). These techniques are far less computational than to re-estimate the new motion vector from scratch but the result is not assured to be the optimum one. Hence, either the compression ratio or the quality will be lessened.

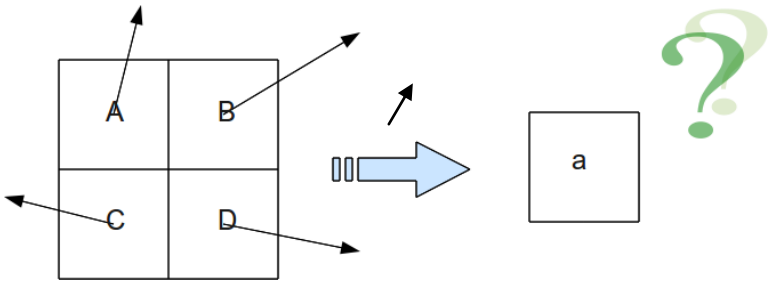


Figure 2- 16 : Downscaling issue example

This key point of video adaptation studies is inspired from the multiple pass encoding process [YU06]. In case of multiple pass encoding, the video encoder analyzes the video many times from the beginning to the end, before encoding process. The encoder extracts and stores information about the original video to a log file. This information is finally used to determine the best solution to compress the video within the constrained bitrate limits and to provide the bast video quality.

The following of this chapter is focused on the overview of different adaptation systems - and their associated techniques - proposed in order to solve this tradeoff between quality and computational complexity. Each adaptation system addresses a unique video adaptation process (either bitrate or spatial resizing or …).

5 Adapatation systems dedicated to bitrate reduction

The most studied adaptation systems are the ones addressing the bitrate reduction between the input video stream and the output one. However, reducing the video bitrate generates lower video quality as shown in Figure 2- 17.

(a) Original Picture

(b) Picture at Half Bitrate

Figure 2- 17 : Pictures at different Bitrate

Changing the video bitrate in realtime has been highly sought in the network community. Indeed, reducing the channel usage is proposed to avoid or reduce network congestion. This is one of the key features for video network management in today's context.

5.1 Low-complexity approaches for bitrate reduction

There are two approaches for the video bitrate reduction. Both approaches aim at increasing the effectiveness of the Zigzag/Run-Level Encoding pair. As explained above, the zigzag process reorders the coefficients of a block in order to optimize trailing zeros. The more trailing zeros there are, the more efficient the Run-Level Encoder is (Figure 2- 18)

5.1.1 Re-Quantification

The first approach used to reduce video bitrate was proposed in [LEI02-1] and [LAV04]. This approach operates by raising the quantification scale of the input video stream. Indeed, increasing the value of the quantization scale helps in generating more null (zeros) quantified coefficients at variable length encoder input. This technique is illustrated on Figure 2- 18, where the quantization scale is doubled between the two blocks. This technique is the most used in bitrate reduction. This technique can be easily controlled but induces blurs in the resulting picture

168	1	-2	1	1	0	0	0
-4	-3	1	-1	0	0	0	0
-2	-3	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

→{168, 1, -4, -2, -3, -2, 1, 1, -3, 0, 0, 0,-1, -1, 1, EOB}

79	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

→{79, 0, -2, -1, -1, -1, 0, 0, -1,EOB}

Figure 2- 18 : Two compression using ZigZag and Run-Level

5.1.2 Frequency Decimation

Frequency decimation ([ASS97] and [SUN96]) is obtained by using low pass filter with a variable threshold. The threshold is modified according to the bitrate aimed also reducing the number of non zeros values, decreasing the number of bits coded in the outgoing stream. Figure 2- 19 shows an example where the first matrix of Figure 2- 18 is decimated to 9 coefficients.Unfortunatly, this technique induces rays in the picture that are more rapidly noticed.

168	1	-2	1	0	0	0	0
-4	-3	1	0	0	0	0	0
-2	-3	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

→{168, 1,-4,-2,-3,-2,1,1,-3,EOB}

Figure 2- 19 : Frequency Decimation example

5.2 Processing chains for bitrate reduction

The aforementioned techniques considerably reduce the complexity of the adaptation system required to realize bitrate adaptation. The presented techniques are seamlessly used in the

simplified adaptation systems presented bellow. In this section, figures depicte adaptation systems. In these figures, bitrate control techniques are implemented in the "*Adaptation*" block.

5.2.1 Open-Loop adaptation system

The open-loop adaptation system ([ASS97]) is adaptation system that is the least computational. Its processing structure is provided in Figure 2- 20. The decoding path is stoped at the inverse quantization step. Coefficient blocks are then adapted using either requantization or frequency decimation techniques. Finally, the encoding path begins at the quantization step. Hence, a lot of computaionnal blocks are avoided. The main drawback of the open-loop system is its high sensibility to errors. This adaptation system do not recalculate residual information of a picture when its reference is modified. Thus mismatchs induced by bitrate adaptation technique are propagated from a frame to the next one. This error propagation is called drift error. Drift errors are stopped only by I frames that do not require reference. Hence the resulting quality of this adaptatation system is highly dependent on the number of I frames in the video.

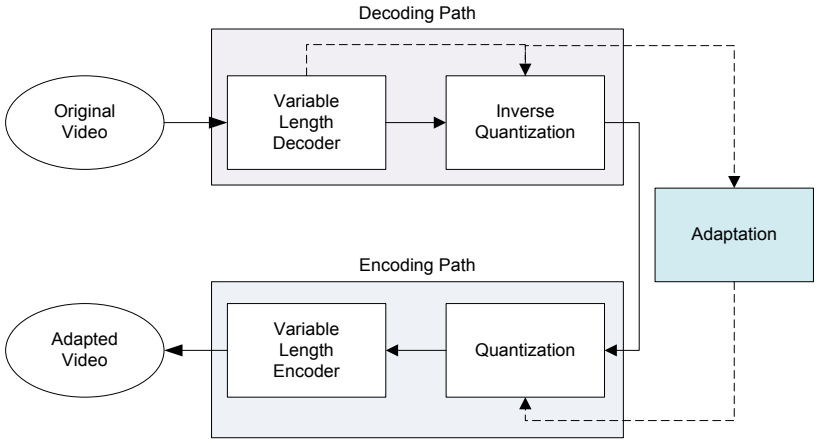


Figure 2- 20 : Open Loop Adaptation System

5.2.2 Simplified Encoder-Decoder (Simplified DCT Domain Transcoder)

Assução and Ghanbari [ASS96] propose a simplified Encoder-Decoder adaptation system that exploit the linearity of the DCT process. The processing chain is provided in Figure 2- 21. This system operates first in the frequency domain and keeps the pixel domain final transformation for a loopback error correction, in order to avoid drift effect.

In this case, the inverse DCT and the motion compensation blocks operate correction processing in both the encoding and decoding path. It is an error correction path that can be merged into a single error correction path to form a refined version of the Simplified Encoder-Decoder system. The modified structure of the system is provided in Figure 2- 22.

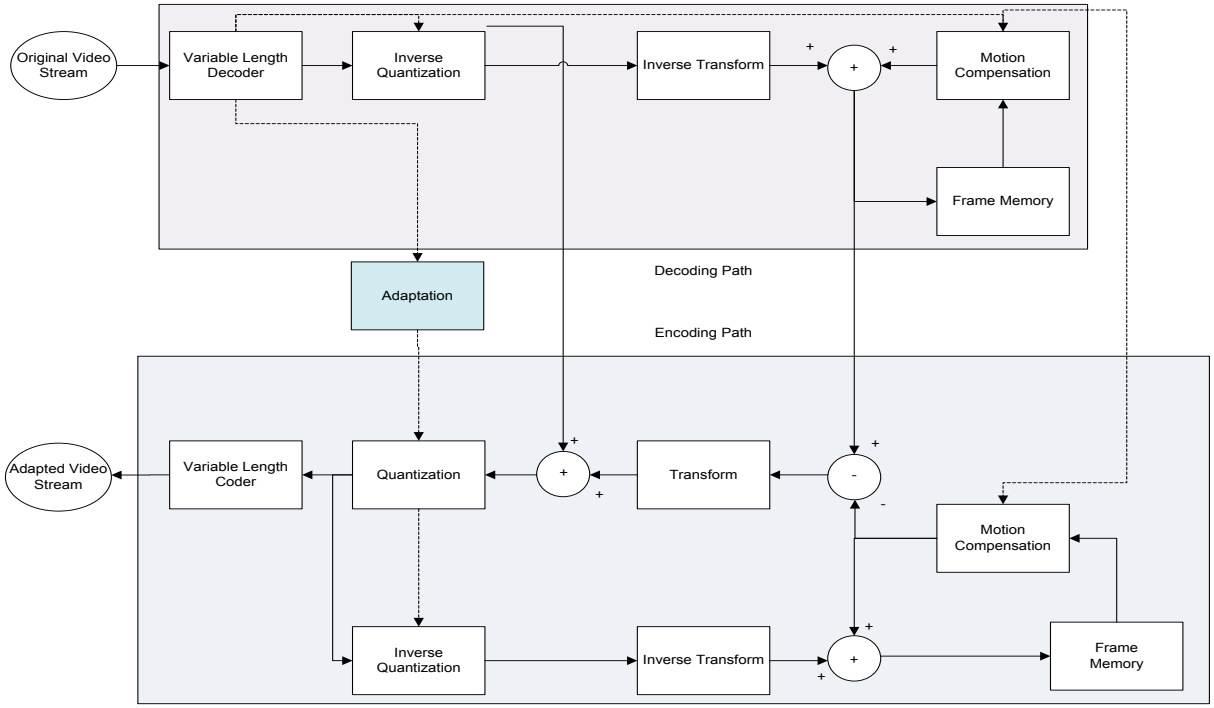


Figure 2- 21 : Simplified Decoder-Encoder intermediate Adaptation System

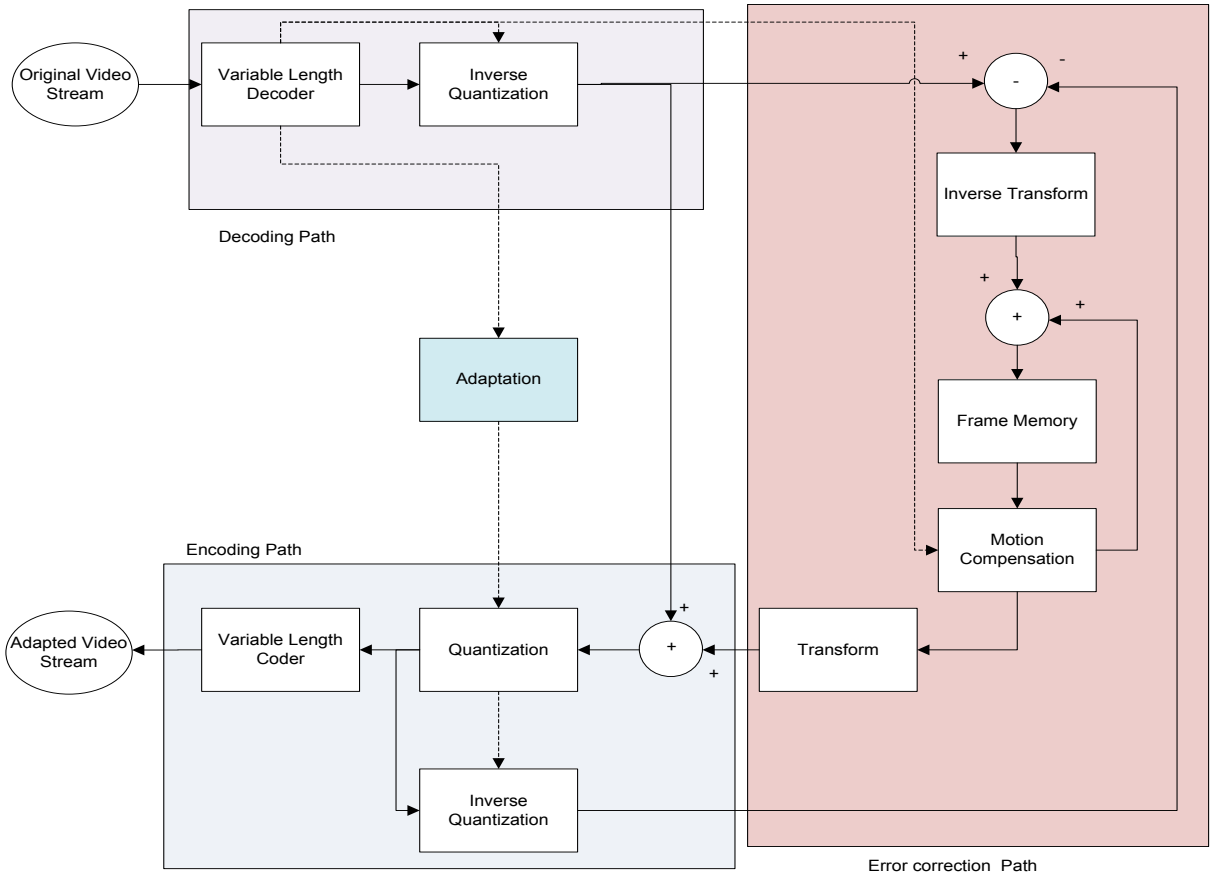


Figure 2- 22 : Simplified Decoder-Encoder Final Adaptation System

This approach removes the drift effect drawback of the open loop. Unfortunately, the quality gain is balanced by a huge increase of the computationnal complexity that the error correction path requires. This systems is based on the linearity of the DCT. Other transforms can be used (wavelet,

integer transform, hadamard …), that may not be linear. Thus this system is not garanteed to overcome bitrate adaptation for a large set of codecs - such as H.264 based codecs that uses integer transform.

6 Adaptation systems dedicated to video downscaling

6.1 Technique for fast spatial resolution adaptations

The spatial resolution video adaptation is one of the most complex adaptation schemes. Indeed, modifying the video dimension requires, for every video frame, an adaptation of:

(c) The pixel information;
(d) The metadata (like the motion vectors).

However, it is also the type of adaptation that has been the most inspiring in terms of information reusability. A lot of solutions ([XIN05], [SHE01]) have been proposed in order to reduce this adaptation complexity.

As seen in a previous example (in Section 3) to halve the spatial resolution of a video, the pixel data have to be recomputed. Moreover, associated metadata, such as the motion vectors, macroblock types, etc. requires recomputations. To avoid recomputations from scratch, it is possible to reuse information obtained from the decoded video stream. For example, the output pixel values and their motion vectors can be obtained by averaging respectively the decoded pixels and motion vectors. However, result quality making such approximation will decrease output video quality. In this section, techniques that are proposed to merge pixels, motion vectors and macroblock type are overviewed. These techniques focus on reusing already decoded information to operate.

6.1.1 Pixel merging techniques

The pixel-merging problem is the most natural. The original video is composed of N*M pixels. The adapted video stream will have N'*M' pixel dimension with N'=N/p and M'=M/p (p is the downscaling factor). The problem associated with video resizing comes from the new pixel value computation. This issue is illustrated in Figure 2- 23. To compute the new pixel value according to the original ones, different techniques providing different quality & complexity tradeoff have been proposed. They are detailed below.

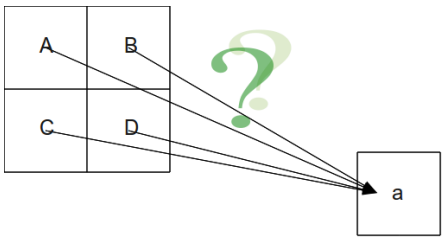


Figure 2- 23 : Pixel Merging

6.1.1.1 Filtering and sub-sampling

This technique interpolates pixel for a reduced resolution by first filtering the original decoded picture horizontaly and verticaly. Then, the filtered pixels are sub-sampled to fit the targeted

resolution. This approach ([SHA00] and [YIN02]) is used to transform pictures that are in the pixel domain. The computation complexity of this technique depends on the filter orders and filter coefficients (value, format).

For example, Shanableh and Ghanbari [SHA00] proposes a 7-tap filter with the following characteristics (-1, 0, 9, 16, 9, 0, -1)/32.

Another approach has been proposed to allow sub-scaling of picture that are in the frequency domain, without cosinus transform requirements. To achieve such interpolation, Yin et al. ([YIN02]) use a frequency domain filter.

From four 8x8 blocks, the filtered block is obtained by processing two 1 dimension frequency filter on both raws and columns. Let A and B be two vectors of size N. The resulting vector E of size N is obtained by equation (2.1)

$$E = f_1 * A + f_2 * B \tag{2.1}$$

Where

$$f_1(k,p) = \sum_{i=0}^{N-1} \varphi_p^N(i) * \varphi_k^{2N}(i) \tag{2.2}$$

$$f_2(k,p) = \sum_{i=0}^{N-1} \varphi_p^N(i) * \varphi_k^{2N}(i+N) \tag{2.3}$$

And

$$\varphi_k^N(i) = \sqrt{\frac{2}{N}}\alpha(k)\cos(\frac{2i+1}{2N}k\pi) \tag{2.4}$$

And $\alpha(k) = 1/\sqrt{2}$ for k = 0, and 1 for k ≠ 0. These filters can be applied in both the horizontal and vertical directions.

6.1.1.2 Pixel averaging

One of the most common approach ([SHA00]) to implement spatial resolution downscaling is to perform a simple pixel averaging to the original N*N pixels. However, this approach is limited to integer scale factor. Moreover, this technique use introduces a blur effect to the resulting picture.

An identical approach was proposed by Lei et al ([LEI02-2]) to perform the same computation in the frequency domain. This proposition aims at avoiding the transformation process in the adaptation system but is linked to the transform domain and thus cannot be used seamlessly by every codec.

6.1.1.3 Discarding high order coefficients

Discarding high order coefficients ([TAN95]) is a technique that operates only in the transform domain. In this case, pixel information are color variations that are stored from the least variable (constant) to the more variable (high frequency) – see section 2.2. The human eye is more sensible to low order coefficient (low frequency) than to high order coefficient. This technique aims at removing the high order coefficients that less impacts the overall quality. As an example, for a half spatial resolution downsizing, the four 8x8 blocks are decimated to become 4x4 blocks (removing the 64-16 = 48 coefficients of the highest order). Then, an inverse 4x4 DCT is computed on each four blocks, in

order to obtain new four 4x4 blocks in the pixel domain that are aggregated into a 8x8 pixel block. (Figure 2- 24)

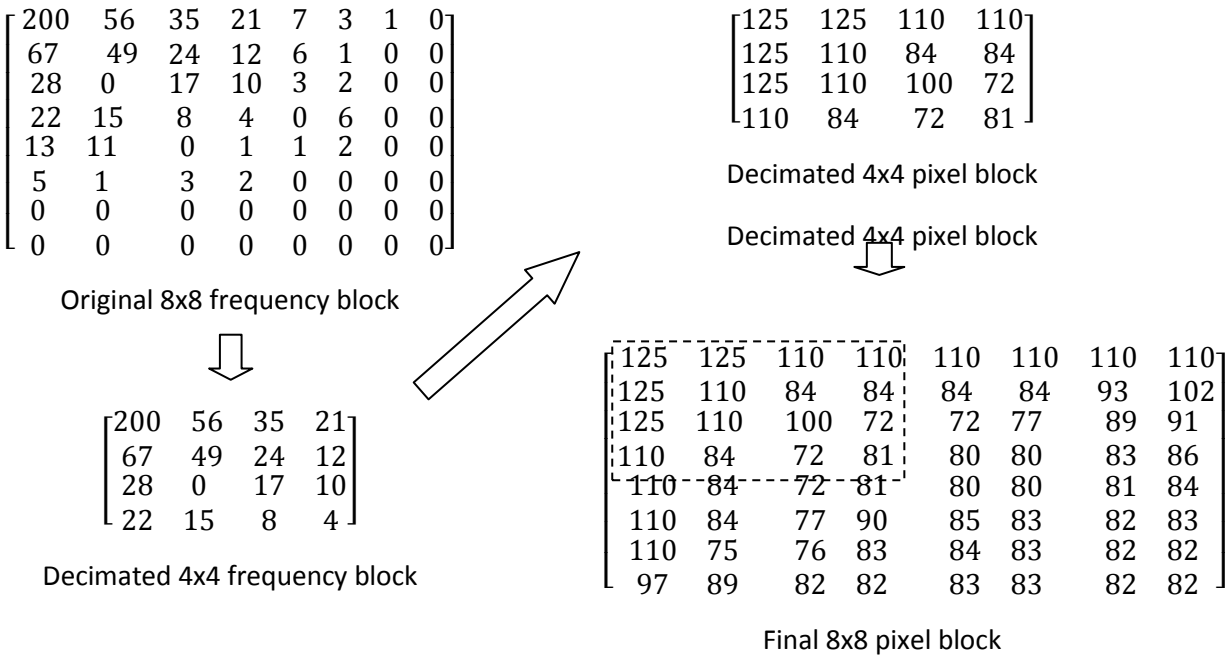


Figure 2- 24 : High Frequency decimation

6.1.2 Motion vector merging techniques

Motion vector manipulation aims at estimate the motion vector that will minimize the residual inforamtation for the new macrobloc using values of original ones. The motion vector estimation is realized without performing the complete motion estimation process ([BEL11] and [ELH11]) from scratch, as already been presented earlier (Figure 2- 16). Five methods providing different tradeoff between efficiency, computation & memory cost were overviewed in [AHM05].

The motion vector of the new macrobloc can be equal to:

- A motion vector from the original ones. Selected motion vector is picked randomly [BJO98];
- An average of a sub-set of the original motion vectors. The motion vector sub-set can be equal to the overall motion vectors [SHE99] or can be equal to the motion vectors that have the same direction [SHA00], or have some correlation between the neighboring blocks;
- A weighted average of the original motion vectors ([YIN00], [SHE97]), where the vector weight depends on the number of non null frequency coefficients of the respectiv blocks, namely its spatial activity;
- A weighted median of the original motion vectors [SHA00];
- The motion vector of the original macrobloc that has the highest DC coefficient [AHM05].

In order to improve the quality of the estimated motion vector, [BJO98] has proposed to refine by a last step of motion estimation on the nearby of the estimated motion vector.

6.1.3 Macroblock type merging techniques

In section 2.3.1, the macroblock type has been presented. There exist frames that are encoded using various macroblock type, as illustrated in Figure 2- 11. The coding mode decision problem arises when heterogeneous macroblock types are present within a single frame. Indeed, if a group of four macroblocks of various types are merged into a unique macroblock, which type shall the resulting macroblock be? This issue is illustrated on Figure 2- 25. This heterogeneity is handled by most video compression standards ([MPEG2], [H264]). Two main approaches have been proposed in the litterature to find a solution to this problem ([XIN02], [BJO98]):

1. If any original macroblock participating in the composition of the new macroblock is of type INTRA, then the remaining macroblock is an INTRA type, else, if at least one original macroblock is of type INTER, then the remaining macroblock is an INTER type, else, every original block are SKIPPED, so the remaining block is coded as a SKIPPED block;
2. In case of heterogeneity, the INTER type prevails, INTRA type macroblocks are seen as INTER with motion vectors reset to zero or predicted from neighbors.

Björk et al ([BJO98]) adds the SKIPPED macroblock which is less dominant than the INTER type but, due to the quantization, INTER macroblock can become SKIPPED. This type is thus verified in the middle of the encoding process.

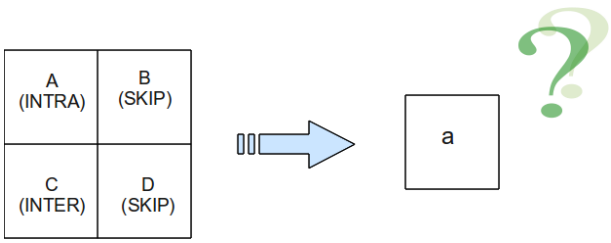


Figure 2- 25 : Macroblock Type Decision

6.1.4 Conclusion on pixel, motion vector and macrobloc merging techniques

Different approaches have been presented to solve macroblock merging issues. They interpolate pixels, motion vectors and macroblock types in order to save computational steps in the encoder path. The three merging techniques – pixel, motion vector and macroblock types – can be chosen independently from one to another. These approaches have their own efficiency. However, the merging technique combination selection and the way they are used, impact the adapted video quality. In the next section, we focus on different systems that are using these MV, Pixel, and type merging algorithms.

6.2 Adaptation systems for spatial video transformation

6.2.1 Open-Loop adaptation system

The open loop adaptation system [YIN02], shown in Figure 2- 26, is very similar to the one previously presented in Figure 2- 20 (related works on bitrate adaptation). Indeed, this processing system without error compensation can be used to address spatial reduction needs. However, the adaptation block that only realize “Quantification Factor”, in Figure 2- 20, is composed, in Figure 2-26, of more processing elements e.g. pixel merging block.

However, this adaptation system that supports video downscaling with a low computational complexity has the same disadvantage that one presented previously: computation errors realized during the adaptation of frame t are propagated to the next adaptated frames {t+1, t+2, etc.} until the next Intra frame happens.

More than inducing drift errors in the adaptation process, the open loop adaptation system is unable to merge macroblocks of different types. In section 2.3.1, page 44, we have explained that frames are not necessarily composed of macroblocks of the same type. This statement has been illustrated in Figure 2- 11. We have tackled that algorithms (section 2.3.1) have been proposed to find the resulting type of a macroblock when operating a 2 to 1 downscaling process. When dealing with macroblocks of type I, data that are encoded are pixel values. When dealing with macroblocks of type P or B, data that are encoded are residual information. In a classic decoder, the adder between the inverse transform and the motion compensation (Figure 2- 15) is responsible for transforming residual information into pixel values that can thus be merged. In the Open loop adaptation system, this adder (and motion compensation) does not exist. Hence, merging residual information with pixel does not make any sense. The open loop adaptation system handles only homogeneous frames (i.e. frame composed of macroblock that has the same type).

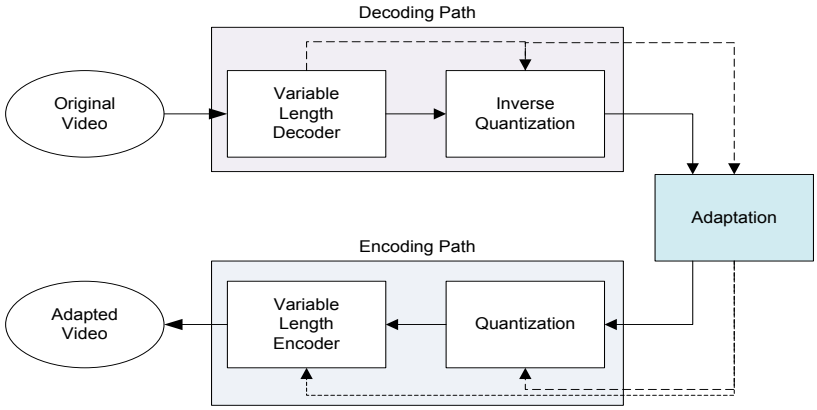


Figure 2- 26 : Open Loop Adaptation System

6.2.2 Drift effect compensation

To remove the drift effect in the open loop adaptation system, an error correction loop was added [YIN02]. This new adaptation system is depicted in Figure 2- 27. The solution is similar to the *Simplified Decoder-Encoder Final Processing Chain* (Figure 2- 22) that operates for bitrate adaptation.

This video resizing solution must provide better quality results inspite of requiering a higher compuation and memory storage.

6.2.3 Drift compensation in original resolution

The “Drift Compensation in Original Resolution” adaptation system ([YIN02]) operates in the same way but includes the down-sampling process inside the encoder part. An up-sample process is used in the error correction loop (Figure 2- 28). This modification allows a better resilience to drift errors than the previous systems. However, the video quality improvement requires higher memory capacity, because the « frame memory » size must store a non adapted video frame instead a downscaled one.

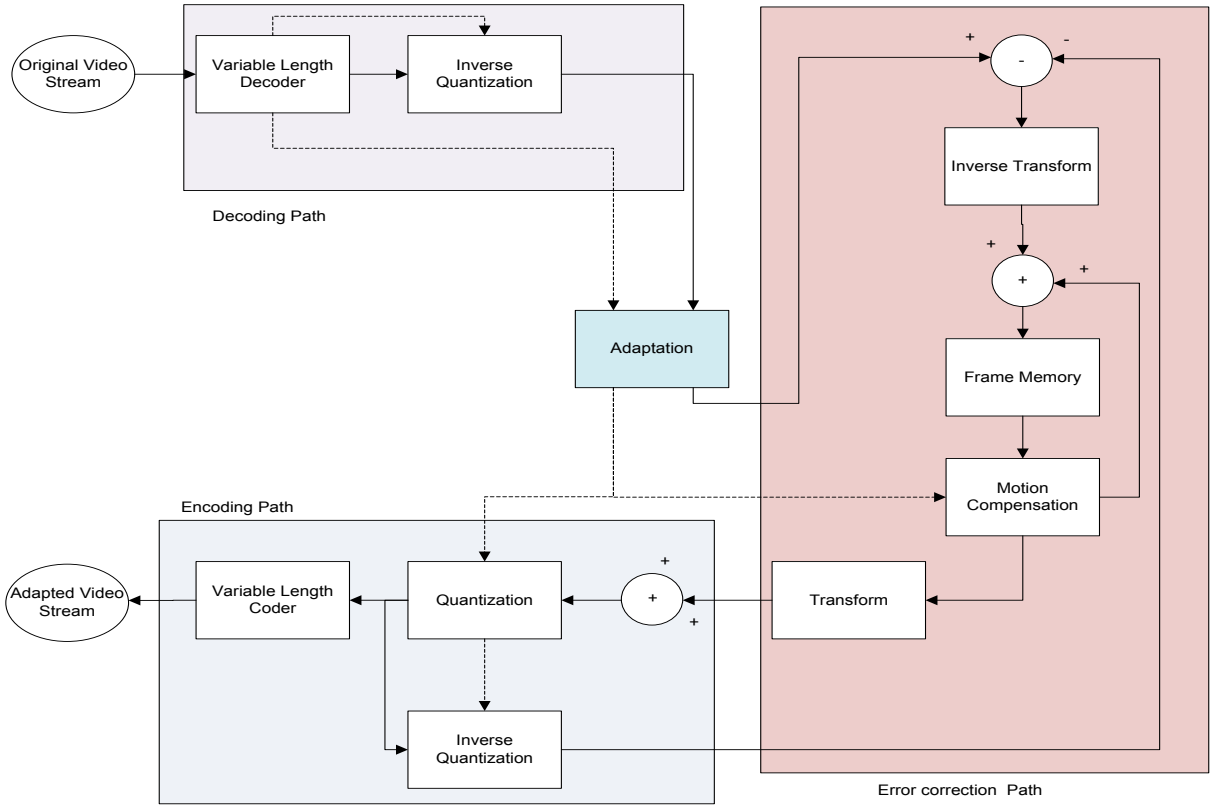


Figure 2- 27 : Drift compensation in reduced resolution Adaptation System

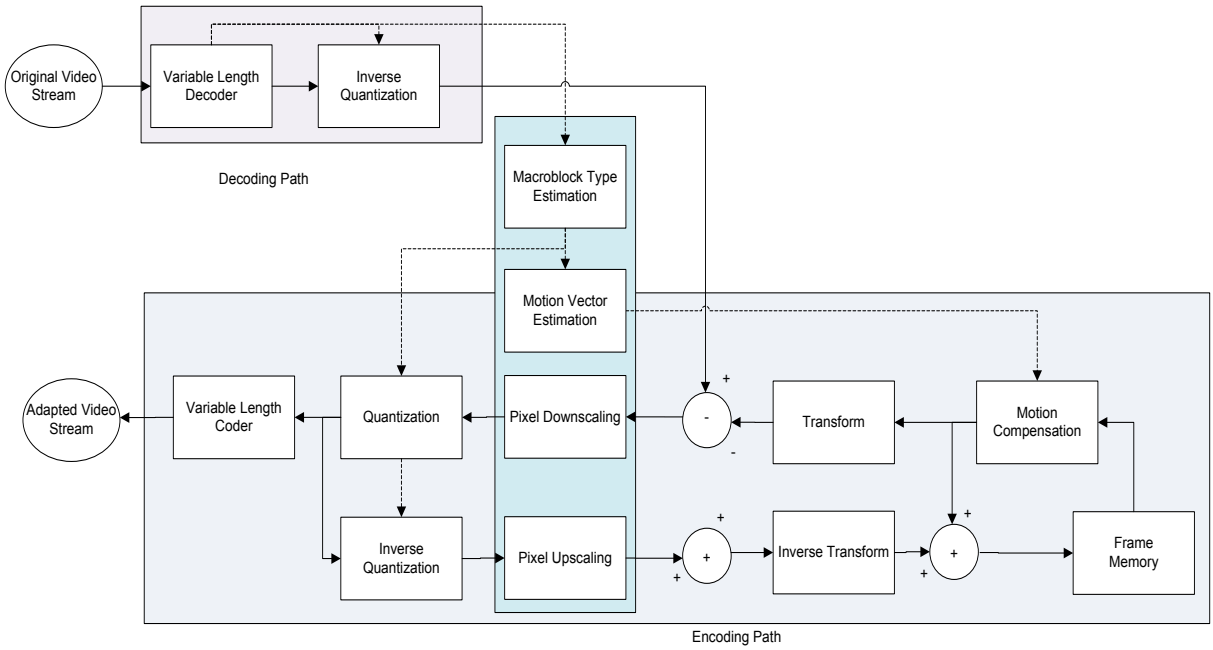


Figure 2- 28 : Drift Compensation in Original Resolution Adaptation System

6.2.4 Partial-Encode adaptation system

The error correction loop (or feed back loop) is highly computational. However, to avoid drift errros, the motion compensation and its associated frame memory (to store reference pictures) are mandatory. Vetro and al. [VET02] considers this issue by proposing the Partial Encode adaptation system (Figure 2- 29)

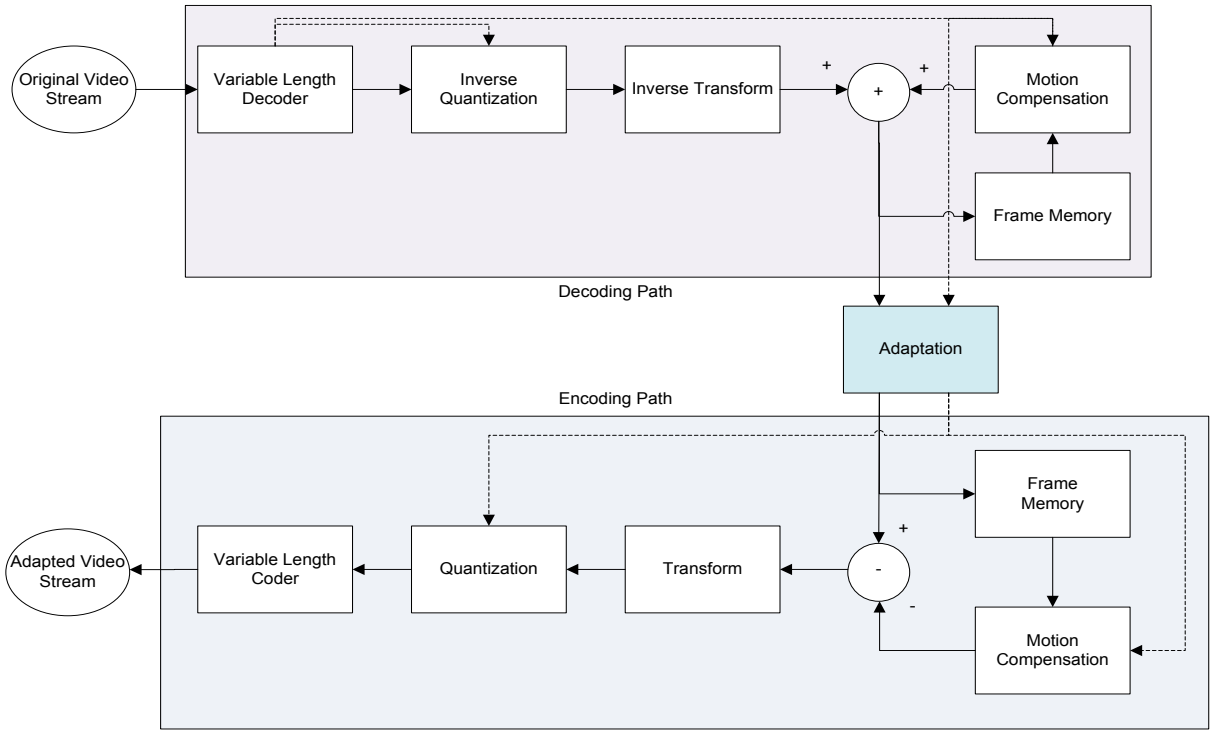


Figure 2- 29 : Partial Encode Adaptation System

This solution offers an interesting tradeoff between the open-loop solution (Figure 2- 26) and the traditional close loop processing chain (Figure 2- 31). Indeed, its video quality performances are better than the ones obtained with the open-loop solution, and its implementation complexity is lower that the traditional close loop one.

6.2.5 Intra_Refresh in Open Loop

The open loop adaptation system can only process homogeneous frame (see section 2.3.1). The intra_refresh adaptation system ([VET02]) has been proposed to overcome this limitation, while being the least computational possible. It derives from an open loop processing chain with a feedback loop used for inter to intra type conversion, when the multiple type macroblock issue appears. Furthermore, the inter to intra conversion imposes intra frame periodically in order to reset the accumulated drift effect. This architecture is highly dependent on the macroblock type decision algorithm employed but realizes a good tradeoff between computational cost and drift effect removal (Figure 2- 30)

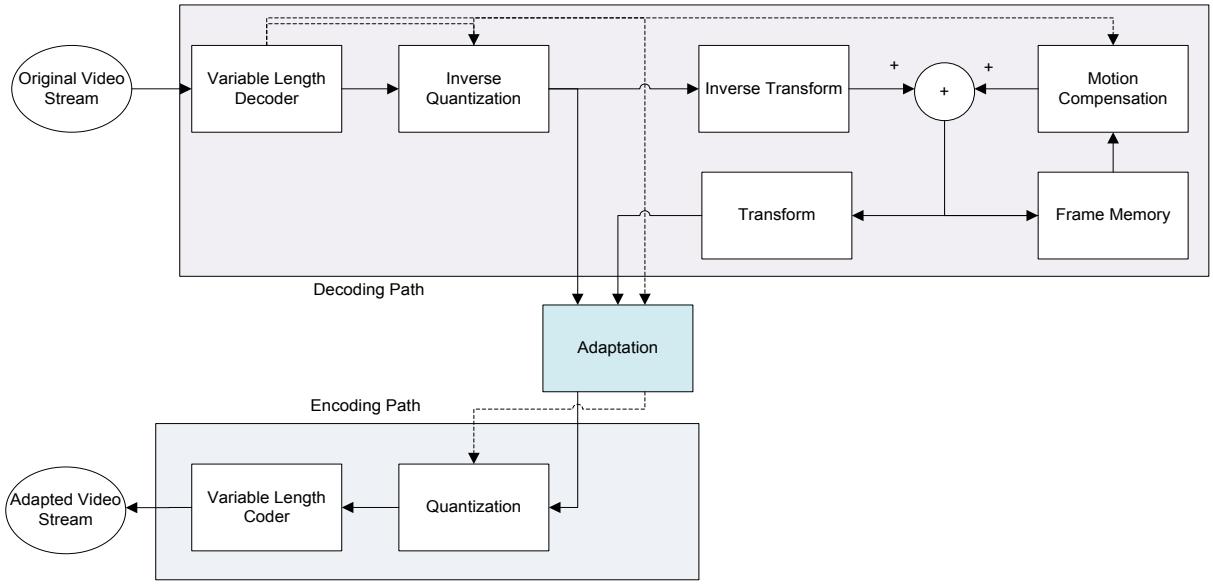


Figure 2- 30 : Intra-Refresh Adaptation System

7 Adaptation systems dedicated to codec change

In chapter 1, the plethora of existing codecs has been illustrated. A complete overview of codec adaptation solutions cannot be done. Let's consider N codecs, for a given codec there are N-1 possible codec changes. Hence the number of codec change systems is N*(N-1). However, the majority of existing codecs follow an international standard or recommendation. Indeed, H.262 and H.264 are widely used throughout the world. The H.262 has been chosen to encode DVD and is the legacy standard for Digital Television. The H.264 has been created as a successor of the H.262 and is twice as efficient as H.262. Thanks to this efficiency, H.264 has led to the arousal of High Definition video standard. This overview is limited to the main issues that arise in standard switching context and their associated solution proposed in the litterature.

7.1 Parameters manipulation

When operating in the pixel domain (using a complete decoder), in order to reduce the computational cost of the adaptation system, the focus is made on translating the incoming standard/codec parameters to the outgoing standard/codec parameters or to estimate the new parameters with the help of the old ones. The main objectives of these works is not to optimize the new codec usage (e.g. using the latest functionalities to improve video stream compression), but to enable a very fast transformation of one format to the other one with similar performances (quality, compression).

Table 2- 2 shows various video parameters for some existing and widely used international standards. Works have been done, focusing on a unique transition (from one codec to another). Most of the issues are relative to the presence of Intra Prediction, frame/macroblock coding type and the different vector block size.

Features	H.261	MPEG-1	MPEG-2	H.263	MPEG-4	H.264	WMV9/VC-1	AVS
Picture Coding Type	I, P	I, P, B	I, P, B	I, P, B	I, P, B	I, P, B	I, P, B	I, P, B
Entropy Coding	VLC	VLC	VLC	VLC, SAC	VLC	UVLC, CAVLC, CABAC	Multiple table VLC	Adaptive VLC
MV Resolution	Int. Pel	½ pel	½ pel	½ pel	¼ pel	¼ pel	¼ pel	¼ pel
Transform	8×8 DCT	8×8 DCT	8×8 DCT	8×8 DCT	8×8 DCT	4×4 & 8×8 Integer	8×8, 8×4, 4×8, 4×4 Integer DCT	8×8 integer
Vector Block Size	16×16	16×16	16×16, 16×8	16×16, 8×8	16×16, 8×8	16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4	16×16, 8×8	16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4
Spatial Intra Prediction	No	No	No	No	No	Yes	No	Yes
Formats Supported	Prog.	Prog.	Prog./Intr.	Prog.	Prog./Intr.	Prog/Intr	Prog/Intr	Prog/Intr
Prediction Modes	Frame	Frame	Field & Frame	Frame	Field & Frame	Field & Frame	Field & Frame	Field & Frame
De-Blocking Filter	In-loop	None	Post	Annex J In-loop	Post	In-loop	In-loop	In-loop

Table 2- 2 : Video standard feature overview[5]

As an example, Feamster et al. ([FEA99]) addresses the MPEG-2 to H.263 transcoding problem with a parameter translation process between the decoder and the encoder parts. They tackled the issues of vector that can or cannot point outside a frame, prediction mode (Field/Frame) and Supported format (such as interleave or progressiv coding). Kalva et al. ([KAL05]) proposed a set of MPEG-2 to H.264 tools to predict H.264 features from MPEG-2 metadata, such as the intra prediction or macroblock mode.

7.2 Transform domain

Some works have been done to remove the transform and inverse transform from the adaptation system in order to reduce its computational complexity. The transform process between two codec can be different – e.g. H.262 and H.264. In this case, there is a need to translate transform domain data from one domain to the other. In the MPEG-2 to H.264 conversion problem, Xin et al. ([XIN04]) propose a converter from the DCT coefficient of the MPEG-2 standard to the integer coefficient of the H.264 standard. This conversion is less computational than a cascaded inverse DCT – integer transform.

8 Adaptation systems supporting different video transformations

In the previous sections we have presented different solutions to realize video adaptation at a lower cost than using the “traditional approach”. These techniques offer tradeoff between video quality and computationnal complexity. However, all of them focus only on a single parameter transformation (bitrate or video dimension or video codec). Other approaches were proposed to address at the same time different parameters adaptation. Three adaptation systems have been thought to handle any kind of adaptation combination [AHM05] and [XIN05]. They are based on the reference processing chain and aim at reducing the overall complexity, while keeping the generality feature of the reference system.

---

[5] J. Golston and A. Rao « Video Compression : System Trade-Offs with H.264, VC-1 and Other Advanced CODECs » White paper, Texas Instrument, August 2006

8.1 Spatial domain adaptation system

This kind of system ([SUN03]) is the first form (Figure 2- 31) of the close loop adaptation system. The name “close loop” is after the feedback loop in the encoding process unlike the open loop adaptation systems (seen in the bitrate adaptation and spatial resolution adaptation sections) that do not possess this feedback loop.

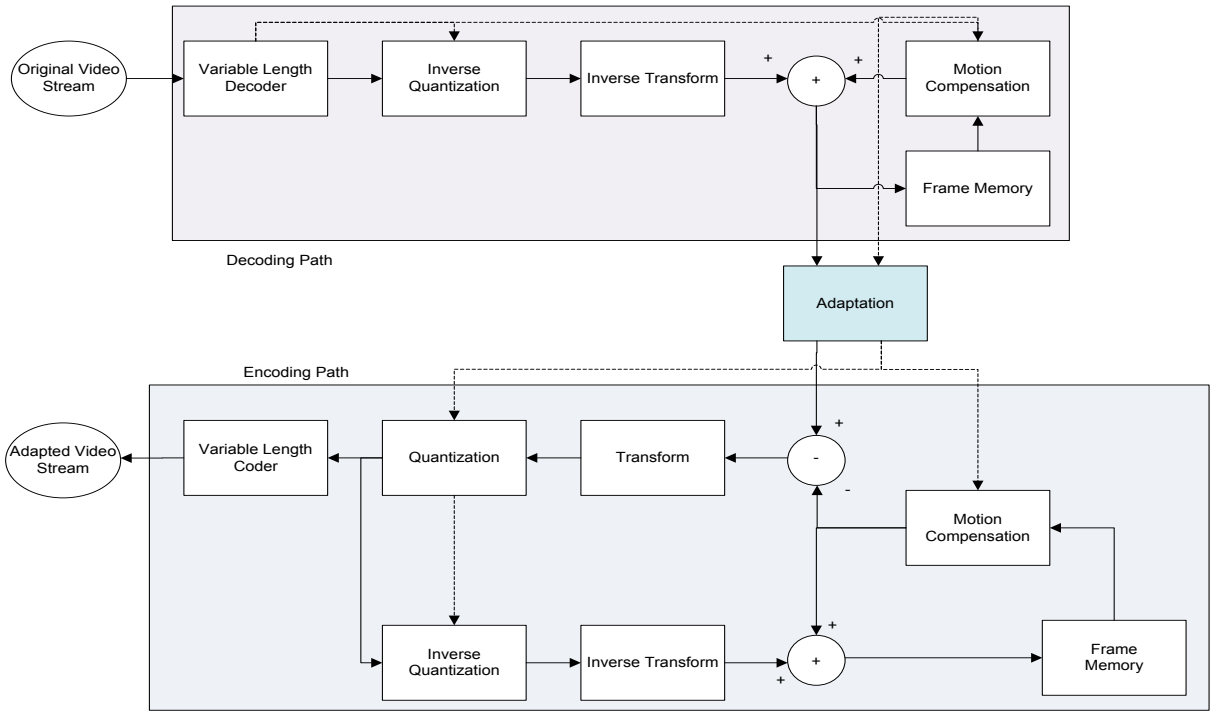


Figure 2- 31 : Close Loop adaptation system

This adaptation system is highly inspired from the reference system. The decision taking engine and the motion estimation process are removed. The adaptation process is in charge of estimating what was decided by the motion estimation (motion vectors, macroblock type), and the decision taking engine (mainly, the quantizer scale). These modifications reduce the encoding processing time by 60-70% ([SHA00]). Shen and al. ([SHE99]) estimate the processing chain to be 37 times less complex when operating on a CIF to QCIF adaptation. This adaptation system achieves a very high quality of video adaptation but is also the most expensive in terms of computational resources compared to the other processing chains.

8.2 Frequency domain adaptation system

The Frequency Domain adaptation system ([ASS98]) is the second form of the Close Loop adaptation system. This system operates in the frequency domain in order to avoid the transform and inverse transform process (Figure 2- 32). Motion compensation that normaly operates in the spatial domain has been modified to directly perform on the frequency domain. This is possible because of the linearity feature and reversible feature of the DCT.

The processing chain keeps a generic behavior assuming that the codec transform is a DCT or, at least, has the same features as the DCT. However, a codec change between two different transform based standards can be handled by a transform domain adaption (see. Codec adaptation considerations in section 7.2).

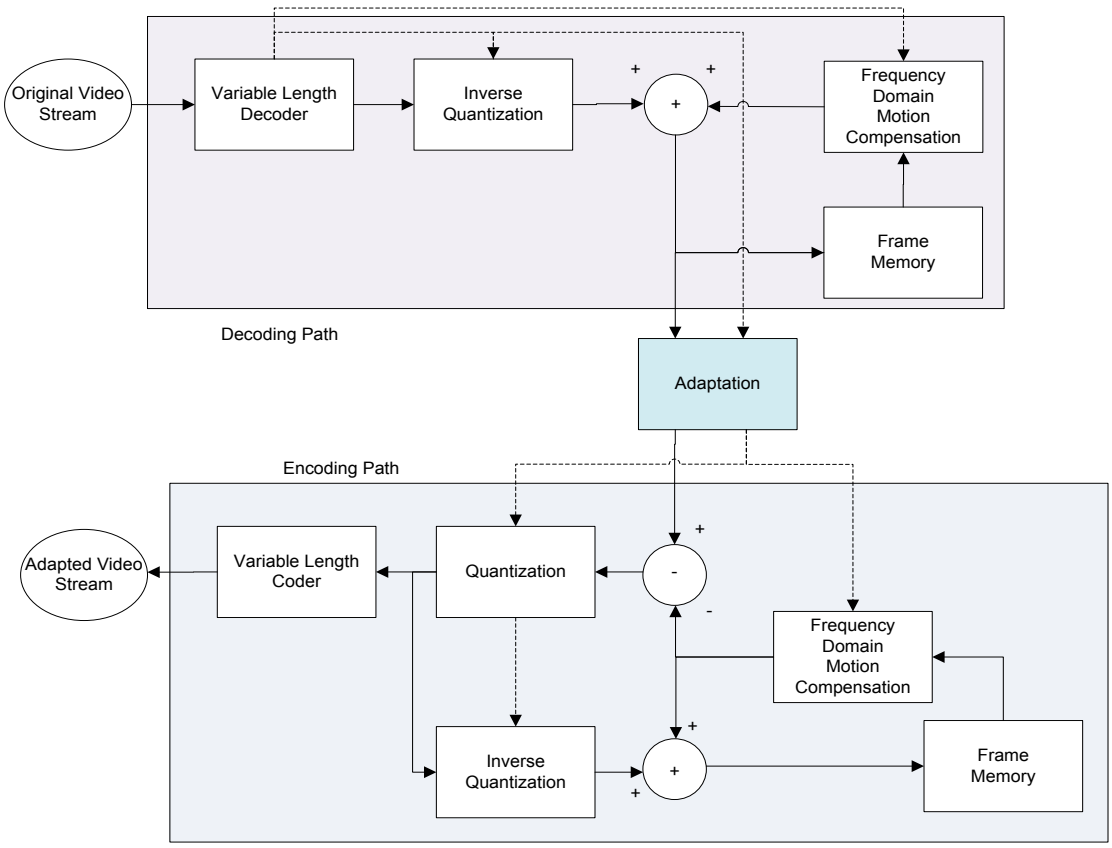


Figure 2- 32 : Frequency Domain Adaptation System

8.3 Hybrid domain adaptation system

The Hybrid Domain adaptation system ([AHM05]) has been designed to tackle to the quality/processing time complexity. This hybrid domain system is a form switching adaptation system. It switches between the spatial domain and the frequency domain adaptation system. The former system needs more time but produces high quality adaptation, the latter is quicker but achieves a poor adaptation. Hence, the hybrid domain system has been designed to minimize drift effect. The spatial domain form is used for P frames that are used as reference and thus need a good adaptation in order to minimize the drift error. I frames stop errors from drifting as they do not use reference and B frames do not spread errors as they are never used as reference. Thus, the frequency domain form is used for I and B frames.

Considering the hardware implementation paradigm, the hybrid domain adaptation system requires both form to be implemented and thus is more expensive than either one form or the other implemented alone. The hybrid domain system achieves a lower quality adaptation than the spatial domain adaptation system. Therefore, in the hardware implementation paradigm, the hybrid domain adaptation system is not considered.

9 Conclusion and adaptation system comparison

Fast adaptation systems have been presented. These systems are all achieving a different tradeoff between video quality and computational complexity. Table 2- 3 summarizes the differences between these systems. Some systems have multiple adaptation capabilities. In order to facilitate the comparison between systems, the adaptation block complexity has not been taken into account.

In chapter 1, the need of a video adaptation system has been presented. In this chapter, video compression techniques have been explained. Those compression techniques are extensively used in the video adaptation framework. Then, an overview of existing video adaptation techniques and systems have been made.

Most of the presented systems are specific to an adaptation process (bitrate change, spatial downscaling or codec change). Furthermore, every tested adaptation system has been evaluated using different testbenchs. Results are not always available and are most likely non uniform. An evaluation has to be done in order to select the appropriate video adaptation system.

REF	Name	Supported adaptation(s)	Presented pages	Computation complexity	Memory requierement
[AHM05]	Reference	ALL	Page 49	+ + + + + + + + + + + + + + + +	+ + +
[ASS97] [YIN02]	Open Loop	Bitrate, (limited Spatial Downsizing)	Pages 54 and 61	+ + + +	
[ASS96]	Simplified Decoder-Encoder (intermediate)	Bitrate	Page 55	+ + + + + + + + + + +	+ +
[ASS96]	Simplified Decoder-Encoder (final)	Bitrate	Page 55	+ + + + + + + +	+ +[6]
[YIN02]	Drift Compensation in Reduced Resolution	Spatial Downscaling	Page 62	+ + + + + + + +	+ +6
[YIN02]	Drift Compensation in Original Resolution	Spatial Downscaling	Page 62	+ + + + + + + + +7	+ +[7]
[VET02]	Partial Encode	Spatial Downscaling	Page 63	+ + + + + + + +	+ +
[VET02]	Intra Refresh	Spatial Downscaling	Page 64	+ + + + + + +	+
[SUN03]	Close Loop	ALL	Page 66	+ + + + + + + + + + +	+ +
[ASS98]	Frequency Domain Close Loop	ALL	Page 67	+ + + + + + + +	+ +
[AHM05]	Hybrid Domain Close Loop	ALL	Page 67	+ + + + + + + + + + + + +	+ +

Table 2- 3 : Adaptation system summary

[6] The illustration of the adaptation system shows only one frame memory but two frames are stored (one for the decoder and one from the feed back loop of the encoder)

[7] This system requieres an upscaling process that has been taken into account

Chapter 3 : A generic video adaptation system

1 Introducing a generic video adaptation system

In the future Internet framework, video adaptation is designed to achieve content– and context–awareness in the global network. This process shall be embedded in low-cost equipment and hence must be the least computational. In the same time, the video adaptation must provide high video quality to guarantee the best user experience.

Video adaptation solutions, provided at the algorithmic level in the literature, have been overviewed in the previous chapter. Those solutions have been designed theoretically, regarding computation complexity and memory complexity. Solution performances have been evaluated using PSNR and MSE metrics that appear to be quite inefficient to fairly compare video quality.

In our study, the constraints set are different. Our objective is to design an embedded system that is able to realize video adaptation in real-time. Original video stream is modified to fulfill the context constraint and must enable the viewing of the original video stream in the best condition, independently of the executed adaptation and its reason. Hence, whatever adaptation is required to match a constraint (network bandwidth, network activity, terminal screen size, etc.), it shall provide the best-perceived video quality.

In our study, the computation complexity is an important constraint because implementation of a real-time system that performs video adaptation is costly and challenging. However, the user experience is the most important constraint: generating adapted video stream with a low-cost system is, at the end, useless, if the video quality is so bad that the user stops watching the video. Due to this acknowledgement, we must ensure that the video quality of the adapted video stream is high enough.

Adaptation system evaluation was realized according to nowadays video characteristics (quality, resolution) and video quality metrics (objective and subjective). Indeed, in literature approaches, proposed systems were evaluated using low frame resolutions (CIF, QCIF) video. As a result, it is difficult to know whether degradation observed in their evaluation are more or less visible in nowadays HD ones. Moreover, commonly used evaluation metrics are the PSNR and MSE. These mathematical metrics were proven as inefficient for video quality comparison [WAN04].

In this chapter, we first present an analysis to select generic adaptation systems from the overview done in the previous chapter. Secondly, two novel adaptation systems are proposed inspired on the previous analysis. Then, we present the metrics used for comparisons and the methodology implemented for the experimentations. Finally, we provide the results and conclude the generic process that best achieves the trade-off between complexity and quality.

1.1 Objectives

The main objective of the adaptation system evaluation is to validate the video quality of the adapted video streams. Indeed, some information are not provided or discussed in the literature:

- The quality information is often an average over time on the video sequence. However, authors never provide information on error distribution over time;

- The quality information is provided using signal processing metrics that are not suitable for video quality comparison (see section 1.3 in this chapter). Indeed, signal degradations (at constant SNR) are not perceived in the same way by humans;
- Video sequences used for adaptation system evaluations were not the same and/or the experimentation conditions or objectives were different. This lack of common methodology makes very hard the comparison of systems.

In order to identify the most interesting solution, the following issues must be solved:

- Evaluation must provide information on video quality obtained by using the adaptation systems;
- Similar experimentation sets for the different adaptation systems may then allow us to compare their performances.

We detail in the next sections the adaptation systems that were selected for evaluation and the methodology used.

1.2 Evaluated adaptation systems

1.2.1 Adaptation system selection

In the previous chapter, a set of adaptation systems has been presented. Based on it, rules to select a generic adaptation system can be:

1. The adaptation system shall work in the pixel domain. The transform domain can vary from one standard to the other (e.g. MPEG-2 to h.264) and a domain converter shall be used for each domain adaptation;
2. A motion compensation system shall exist in the decoding path for the adaptation system to be able to handle various macroblock types in one frame. Like the intra refresh adaptation system (see chapter 2 section 6.2.5), the motion compensation is sufficient on key macroblocks.

Concerning spatial resolution adaptation, it requires modification of a lot of information (pixel, mv, macroblock type …). On the opposite, bitrate adaptation is a simple process. Hence, in our evaluation, we have only considered adaptation system capable of spatial resolution adaptation that follows the two aforementioned rules. The three adaptation systems that have been selected are:

1. The "***close loop adaptation system***" – This system presented in chapter 2 (Figure 2- 31, section 8.1) is the most complex one. Indeed, it is composed of most of the encoder and decoder processing elements. The advantage is that it is able to support the overall adaptation use cases (resolution, bitrate, codec);
2. The "***generic partial encode adaptation system***" – This system has been extended from the partial encode processing system (Figure 2- 29). It was developed to resolve only the video downsizing issue (see chapter 2 section 6). To authorize such adaptation system to perform bitrate reduction, some processing has been added for this usage, as presented in Figure 3-1. This adaptation system supports codec adaptation because it works in the pixel domain (RAW format for the pictures);
3. The "*generic intra refresh adaptation system*" – This system presented in Figure 3- 2 is the less complex one. It is inspired from the intra refresh adaptation system, which is operating in the transform domain and thus does not follow the first rule. However, by moving the transform and inverse transform at key places in the system, the new adaptation system can be selected.

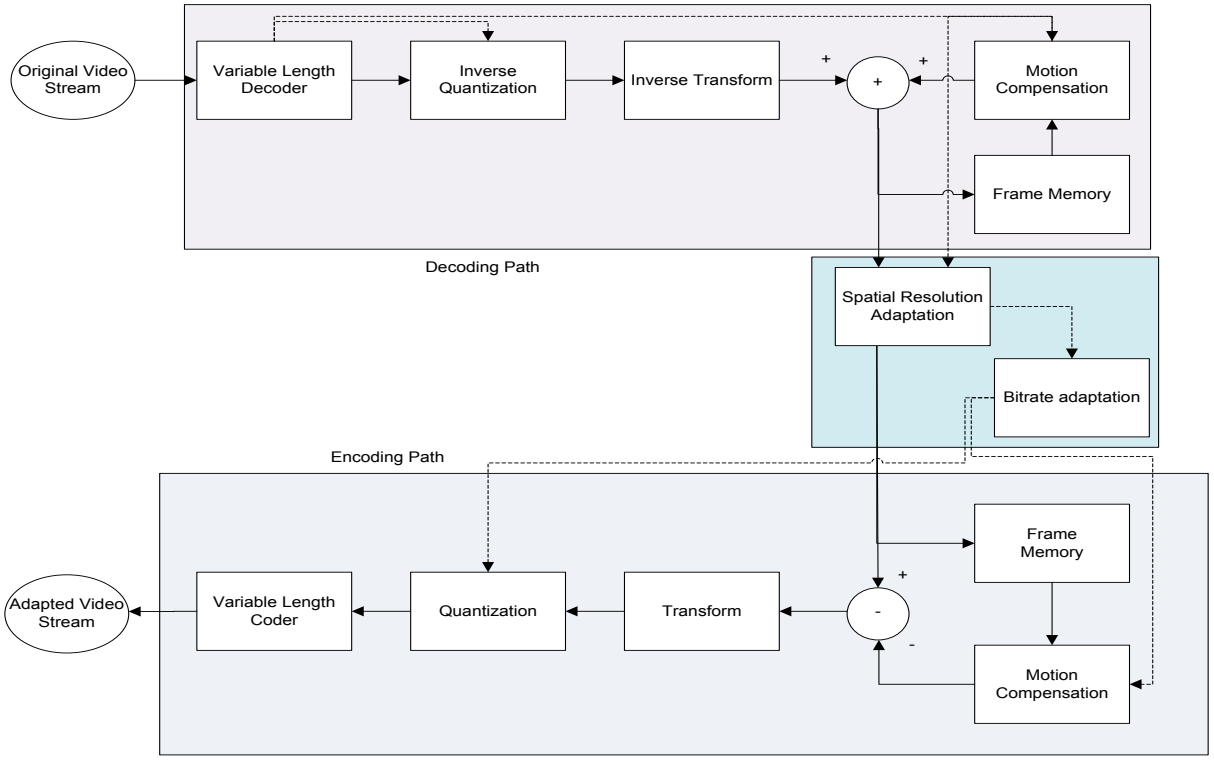


Figure 3- 1 : "*Generic Partial Encode adaptation system*"

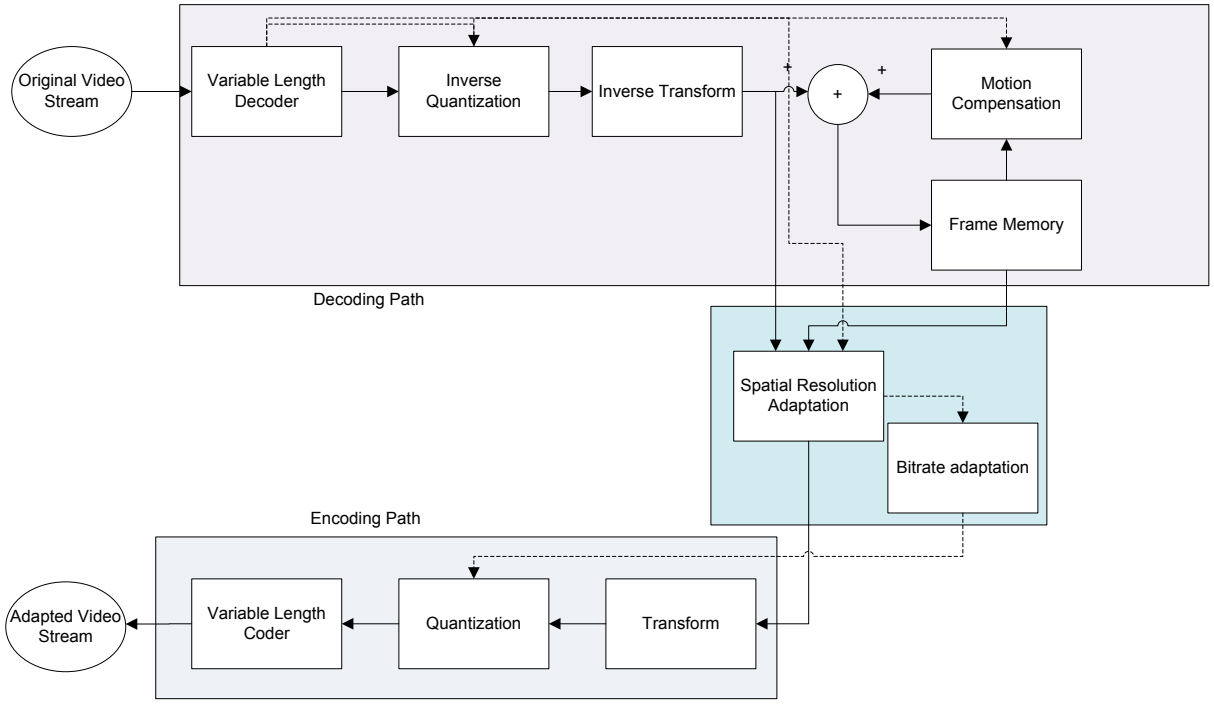


Figure 3- 2 : "Generic Intra Refresh adaptation system**"**

The other adaptation systems presented in chapter 2 were discarded of the performance evaluation because they do not respect the two rules. Indeed, it has been shown in [BJO98] and [LEI02-2] that the adaptation systems working in the frequency domain such as Simplified DCT Domain Transcoder(Figure 2- 22), Frequency Domain Transcoder (Figure 2- 32) and Intra_refresh processing chain (Figure 2- 30), are not suited for video codec adaptation. Indeed, in order to realize the

transformation from one standard to another one (for example, MPEG-2 and H.264), pixel data must be transformed in a shared representation format. Commonly used video standard do not share the same frequency domain format.

1.2.2 Adaptation (downscaling) module design

Many techniques have been proposed in the literature (see section 6.1 in chapter 2) to implement the downscaling module. Indeed, video downscaling requires the computation of the (1) new pixel values, (2) motion vectors and (3) macroblock type (I, B, P).

Because these pixel and motion vector computations impact both the adapted video quality and the computation/memory complexity, it is necessary to evaluate them precisely. Moreover, in order to assert the performances of the evaluated adaptation system, different resizing algorithms will be tested for each adaptation system. This way, we can ensure that if an adaptation system is better than another, it is not because of a lucky good match between the adaptation system and the downscaling algorithm (motion, pixel).

Table 3- 1 provides the configuration that has been evaluated for each adaptation system. The configuration number will be used in the "experimental part" of this chapter to simplify the analysis of the quality charts.

Number	Pixel adaptation technique	Motion vectors adaptation techniques
1	Linear	Random
2	Linear	Quadrant
3	Linear	DC Max
4	Linear	Mean
5	Linear	Weighted Average
6	Quadratic	Random
7	Quadratic	Quadrant
8	Quadratic	DC Max
9	Quadratic	Mean
10	Quadratic	Weighted Average

Table 3- 1 : Downscaling Technique Classification

1.3 Evaluation metrics for video quality

In signal processing, one of the main aims is to keep the signal quality as high as possible. Thus, quality metrics have been proposed to rate the quality of the signal and hence the performance of the processing engine.

1.3.1 Mean Squared Error (MSE)

MSE is the first metric historically adopted in image processing to evaluate the performance of the video or picture transformation such as compression. The mean square error between the two pictures is thus defined as:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - I'(i,j)]^2 \qquad (3.1)$$

Where m and n are the width and heigh of the pictures, I is the original picture and I' the rated picture.

This commonly used video quality metric has a major issue: obtained values do not really represent the video quality loss from a human point of view. This assertion is demonstrated in Figure 3- 3where the 6 pictures have the same MSE value.

As demonstrated through Figure 3- 3, this metric is not reliable to measure the impact of a distortion on pictures. The image quality experienced by the End-User is not similar, whereas the MSE is the same. Further consideration about the use of MSE to evaluate image quality is discussed in [WAN09]

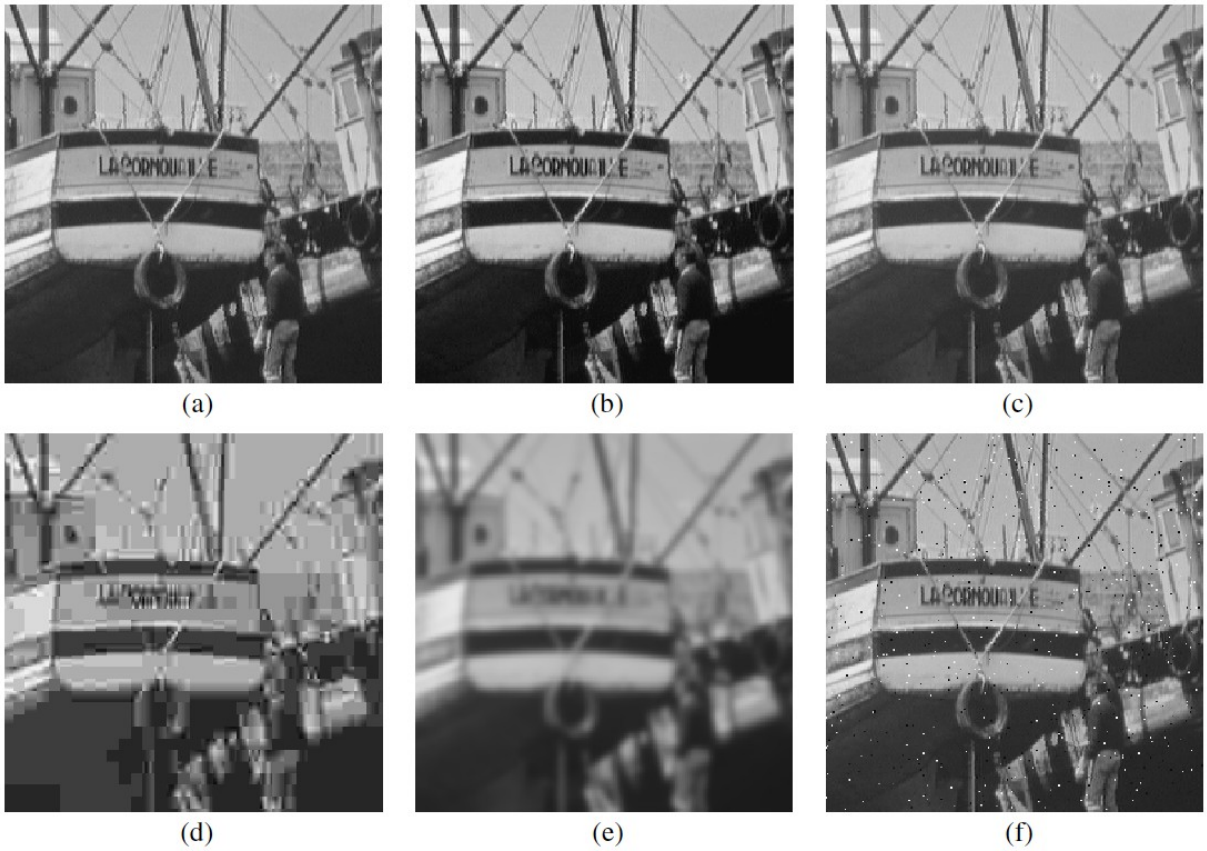


Figure 3- 3 : Comparison of "Boat" images with different types of distortions, all have MSE=210

1.3.2 Peak signal to Noise Ratio (PSNR)

Signal processing widely uses objective metrics that can be automatically computed. The most used is the Peak Signal to Noise Ratio (PSNR) that rates the ratio between the maximum signal and the noise, in a logarithmic way (Equation 3.2). The peak signal is obtained by the maximum possible pixel value in the image.

$$PSNR = 10 \times \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{3.2}$$

Where MSE is computed using Equation 3.1 and MAX_I is the maximum pixel value possible int the original I image.

The reliability of this evaluation metric is quite similar to the MSE one. Indeed, it suffers the same drawback: PSNR value does not represent the real picture quality/degradation detected by human

eye. An example of this drawback is illustrated in Figure 3- 4. Figure 3- 4 (b) and (c) correspond respectively to Figure 3- 4 (a) with a brightness shift and an impulsive noise addition.

Figure 3- 4 (b) and (c) are identical to Figure 3- 4 (a) (they are damaged). Considering a PSNR evaluation, Figure 3- 4 (b) has a lower quality (lower PSNR) than Figure 3- 4 (c). However, from a human point of view, it is the opposite: The bright shifted Figure 3- 4 (b) looks better than the noisy Figure 3- 4 (c).

Figure 3- 4 : PSNR evaluated pictures (a) original picture (b) brightness shift PSNR=18dB (c) inpulsive noise = 12dB

1.3.3 Structural Similarities (SSIM)

PSNR and MSE are statistical oriented quality metrics used for signal processing technique evaluation. These metrics do not consider the content the signal carries. Although, a poor PSNR or a high MSE value for a video will for sure estimate a poor perceived quality, two pictures or video with the same PSNR/MSE can definitely be of different perceived qualities. This is why many research studies have emerged in the field of video quality evaluation metrics.

Among the studies, Wang and al. [WAN04] proposed a metric that asserts a quality rate to a picture by comparing its structural similarities against the reference one. This metric is named the Structural Similarity (SSIM) and its computation is provided in Equation 3.3.

$$SSIM(x,y) = \frac{(2\,\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3.3)$$

Where:

- μ_x is the average of pixels in x;
- μ_y is the average of pixels in y;
- σ_x is the variance of pixels in x;
- σ_y is the variance of pixels in y;
- σ_{xy} is the covariance of pixels in x and y;
- c_1 and c_2 are small variable to stabilize the division

This metric that is Human Visual System Oriented has been proven to be more effective to judge picture quality. However, its computation complexity is higher. Pictures show in Figure 3- 5 illustrate that this metric is more reliable that other ones to measure the human perception of the picture quality.

Figure 3- 5 : Various picture transformations[8] and their associated SSIM values

1.3.4 Conclusion

To fairly compare the adaptation system performances considering human perception consideration, we decided to use SSIM as our metric to evaluate the visual quality obtained by each adaptation system. Videos are composed of trailing frames. Hence, we use the mean SSIM of every frame composing the video to evaluate the video quality and thus the video adaptation process. However, standard deviation and SSIM distribution over time are also shown to strengthen our study.

1.4 Video set used for quality comparison

In order to fairly compare the performances of different adaptation systems, a set of video streams has been selected. Videos have been selected based on activities such as background (moving or not), number of moving object and their speed, etc. All these differences help in fair comparison of adaptation system (and internal techniques) because each adaptation system or technique may be better at adapting a kind of video (e.g. fast motion) and not quite as good for other video (e.g. slow motion).

All the video streams used for evaluation have been recorded directly from digital TV channels (DVB-T) to emulate real life conditions. Table 3- 2 lists the tested videos and their characteristics.

Video Name	Characteristics	Video dimension	# of pictures
Candidat	Portrait; slow motion	720x576	247
CITY	Slow traveling	704x576	497
CREW	Lots of slow movement	704x576	497
Harbour	Movement in Background	704x576	597

[8] https://ece.uwaterloo.ca/~z70wang/publications/SPM09_figures.pdf

Presentatrice	Sill background, slow motion	720x576	247
SIMPSON	Various fast movement	720x576	372
SOCCER	Various moving object	704x576	597
Tennis	Fast moving object on still background	720x576	247

Table 3- 2 : Characteristics of Test Videos

2 Adaptation systems evaluations

2.1 Introduction

In order to identify the best adaptation system according to the video quality constraint, we have re-evaluated the literature approaches: using a human visual system-oriented metric (SSIM) and exploring the different adaptation algorithms provided in the literature. The adaptation systems have been evaluated considering two use cases:

1. Resolution change: Firstly, the video dimension of the input video stream is downscaled to fulfill the terminal display size. The selected downscaling factor is set to 2;
2. Bitrate change: Secondly, the video bitrate of the input video stream is reduced to fulfill the network available bandwidth.

In these two use cases, the input and output codecs are the same.

To realize video quality comparisons, models of the adaptation systems have been developed in SystemC. These models consume real video streams from hard drive and generate adapted video stream that are saved for quality evaluation. SystemC models are generic enough to support the different couples (pixel, motion vector) techniques presented in Table 3- 1.

2.2 Video downscaling evaluations

2.2.1 Quality evaluation system

The first adaptation system evaluation focuses on the real-time video downscaling use case. Every adaptation systems are evaluated in an identical way as presented in Figure 3- 6.

This quality evaluation approach is based on: (1) the SystemC models that we have developed and (2) on a Quality Evaluation Tool (QETool) that was also implemented for this purpose. QETool compares two video streams and provides SSIM values for each frame of the video. The SSIM value for a video is the average of all its frame SSIM values.

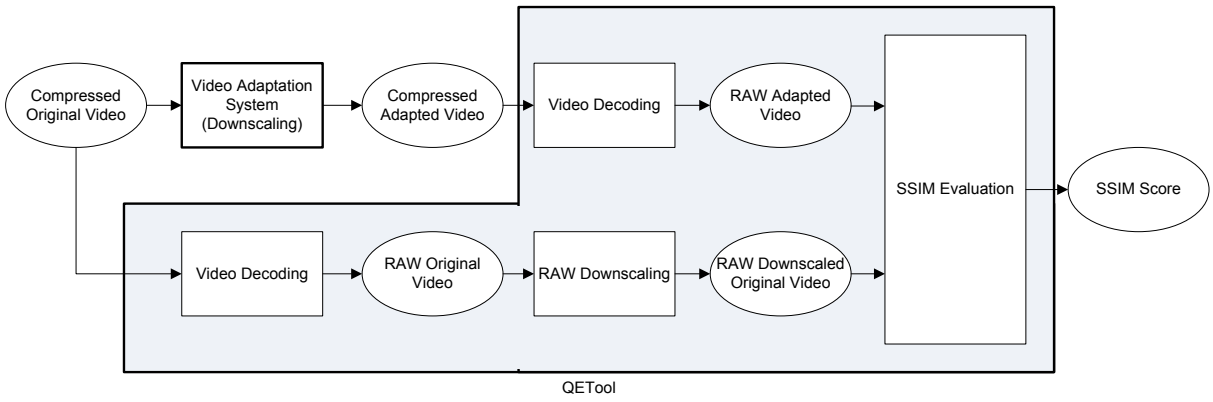


Figure 3- 6 : Quality Evaluation System

The quality evaluation process is based on three main steps:

1. The original video stream is decompressed, downscaled by a factor 2 and then recompressed using the same video codec. These transformations are realized by the SystemC model of the adaptation system currently evaluated. The adapted video stream is stored on the hard drive;
2. QETool converts the original and the adapted video streams to RAW format for comparison purpose. However, dimensions of video are not the same. Original video in RAW format is downscaled by factor 2 to respect the downscaling use-case;
3. The RAW video having the same dimensions are then compared, the quality metrics are computed and information required for comparison are saved.

In the following sections, we provide the quality results obtained for the three adaptation systems under evaluation.

2.2.2 Performance evaluation of the "close loop adaptation system"

Quality results of evaluated videos using the "***closed loop adaptation***" system with the ten technique set (depicted in Table 3- 1) of resizing techniques are displayed in Figure 3- 7. SSIM results provided in Figure 3- 7 show that the video quality measured at the terminal side are excellent. Indeed, the SSIM value is never lower than 95%. This result indicates that the adapted video, once displayed, is very similar to the original one, once decompressed and resized by the terminal device.

Moreover, Figure 3- 7 demonstrates that the pixel and the motion vector merging techniques have a very low impact on the adapted video quality. Indeed, the video quality of the adapted videos varies from +/- 0.06% depending on the technique pair (for the worse case: Presentatrice).

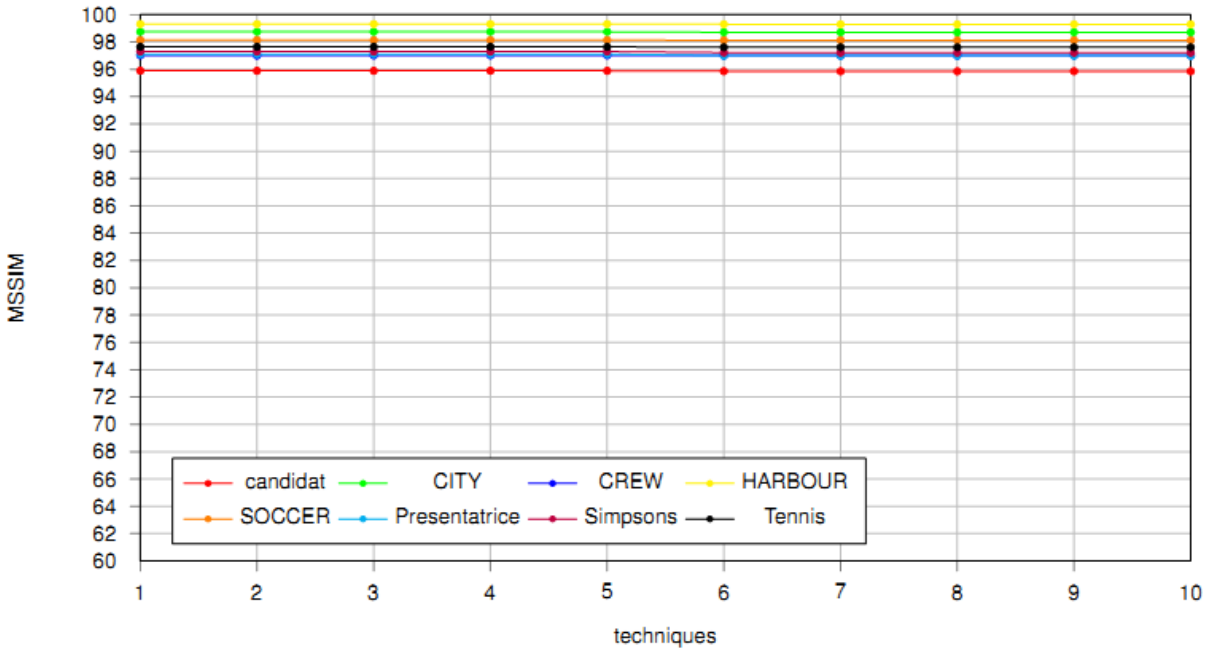


Figure 3- 7 : SSIM Result for "*Closed Loop adaptation system*"

To compare video quality (and thus, video adaptation system quality), video bitrate has to be taken into account. Indeed, the same video with different bitrate may have different quality for the video with the higher bitrate contains more information and thus more quality. Hence, compression ratio

(i.e. bitrate compression) shall be measured along with quality results for a fair comparison. Figure 3-8 provides compression ratio measured between the original video and the adapted ones. Figure 3- 8 demonstrates that even if the video quality achieved by different adaptation techniques is practically the same (Figure 3- 7), techniques impact on the compression efficiency of video streams.

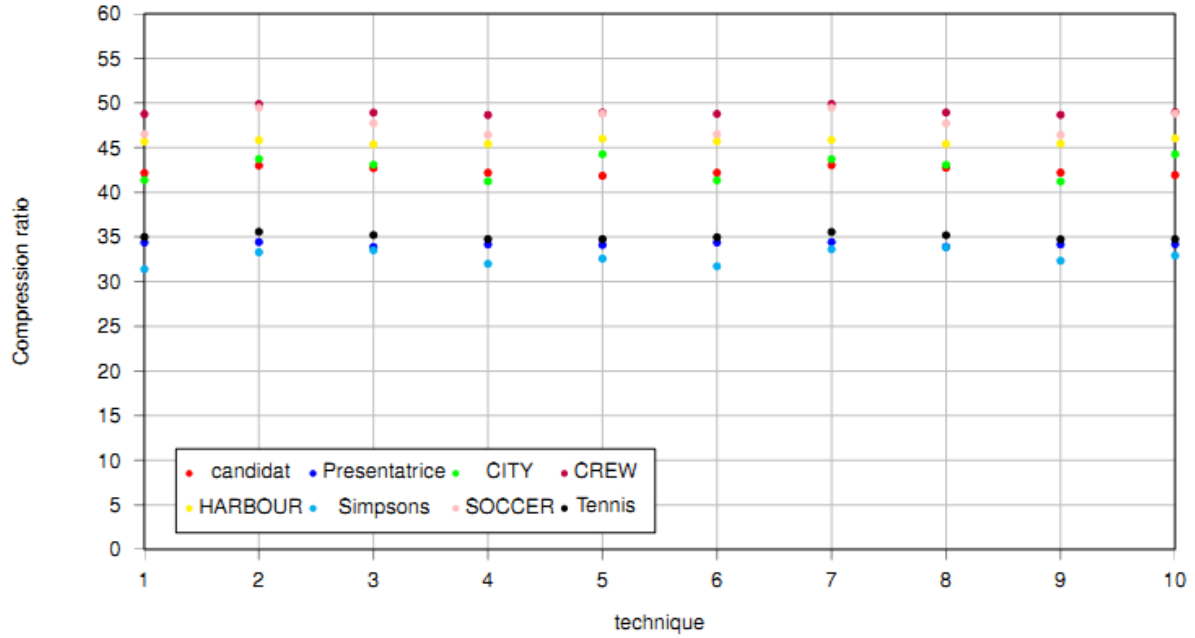


Figure 3- 8 : "*Close Loop system*" compression ratio

These different ratios come from a more or less good match between estimated pixels and estimated motion vectors. A good match between motion vector and pixels means fewer differences between the predicted frame and the actual frame. The fewer the difference, the more compress the video can be (the lower the bitrate). On the contrary, a poor match between motion vectors and pixels means a lot of differences between the predicted frame and the actual frame which result in either a higher bitrate (to store the differences) or a lower quality (when getting rid of the differences).

The least efficient approaches are number 1 and 6; they correspond to the "random merging" technique for motion vectors. Techniques 1 and 6 provide quality equivalent videos however, they require higher video bitrate to do so. Random choice for motion vector creates more mismatch than carefully chosen motion vector. As a result, techniques 1 and 6 generate higher residuals during the compression steps, these residuals are costly to store. Otherwise, the techniques achieving the best compression ratio are techniques 2 and 7. Both use quadrant technique to estimate motion vectors. The worst compression ratio is achieved by the weighted average motion estimation technique (techniques 4 and 9). But the maximum compression ratio difference among techniques is roughly 3% which is not high enough to declare which technique is the best.

Downscaling video dimension by a factor 2 reduces the amount of RAW information (pixels) by a factor 4. A video bitrate reduction of 75% is obtained on uncompressed RAM data. However, bitrate reduction on the compressed video stream is lower. This difference can be explained by different factors:

- Metadata set in the compressed video stream is not reduced by 4, i.e. the slice tags that are inserted to delimit video lines are reduced only by two;
- Some compression optimizations performed during the original video encoding loose effectiveness after the downscaling process, e.g. some video macroblocks that have precise motion vector can sometimes be skipped (their residuals are not stored in the compressed video stream); however, after video downscaling this opportunity cannot be realized anymore;

- Picture structure changes due to the downscaling process. In the Predicted frames, depending on the picture content, some macroblocks can be Intra-coded. These Intra blocks are costly but they improve video quality. During the downscaling process, these Intra blocks contaminate their Predicted (low cost) neighbors. This contamination reduces drastically the compression efficiency in pictures that have at the same time Intra and Predicted blocs.

In the considered use case, the bitstream size of the adapted video is not an issue. The most important parameter is the video quality. According to conclusions from Figure 3- 7, this processing chain is efficient for video downscaling whatever the pixel and the motion vector merging technique used.

2.2.3 Performance evaluation of the "generic partial encode" adaptation system

The same evaluation has been realized on the "***generic partial encode adaptation system***". This adaptation system looks like the previous one. The main difference comes from the fact that the "closed loop" system decompresses the adapted video stream with a feedback loop and compensates the computation (pixel and motion vector merging) errors using already encoded frames. This feedback loop emulates the decoder behavior in order to minimize differences between the encoder and decoder references.

This feedback loop does not exist in the "***generic partial encode system***" to save computational cost (around 1/5th of the total computational cost). The frame used as reference is from the decoding path but not from the feedback loop. There is no emulation of the final decoder (the one located at the terminal side) and thus there exist differences between reference frames used by the encoder path and those used by the final decoder which will lead to small quality loss.

Figure 3- 9 provides the results of the SSIM evaluation. This second set of results shows that the adapted video is also very close to the original one. Indeed, this assertion is proved by high SSIM scores for the video set. Scores are close to 100% and vary from 95% up to 99%.

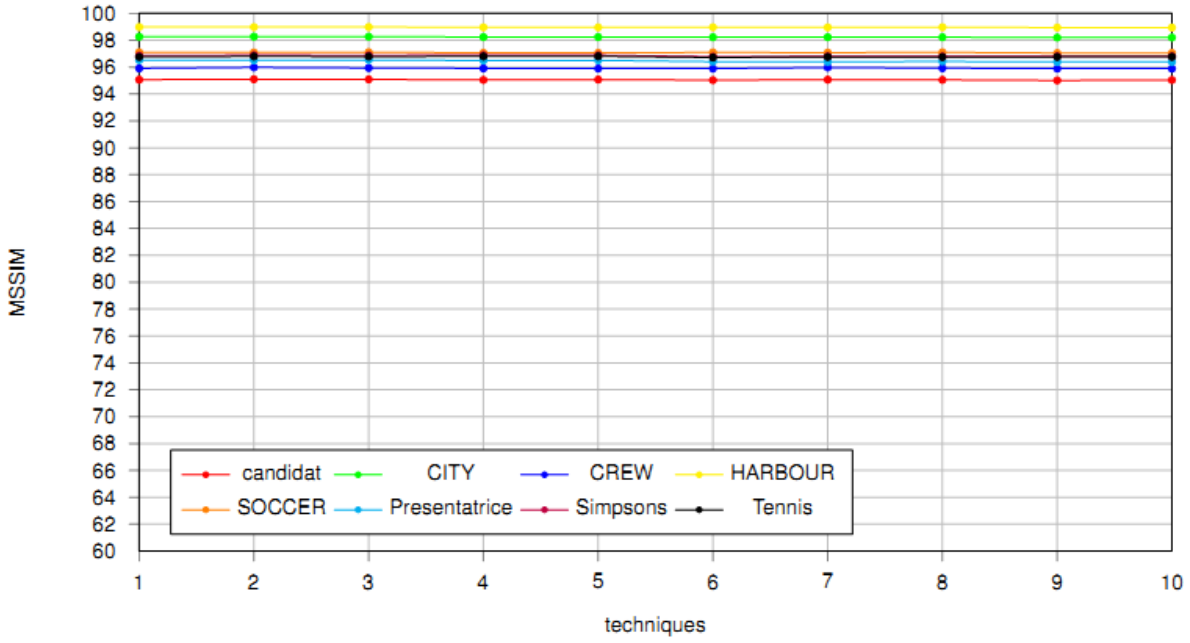


Figure 3- 9 : Result SSIM for "Partial Encode adaptation system"

Quality results presented in Figure 3- 9 compared to "***closed loop***" ones (Figure 3- 7) show that this adaptation system provides lower quality videos due to this encoder-decoder mismatch. However, the quality gap is low. For example, in our evaluation, the average difference of SSIM values between the "***close loop***" scores and the partial encode scores is 0.70 (up to 1.12 all video and technique pair taken together).

The compression ratio of an adapted stream compared to an original one is provided inFigure 3- 10. As stated for the close loop results, high video quality provided by the adaptation system is put in balance with bitrate increase.

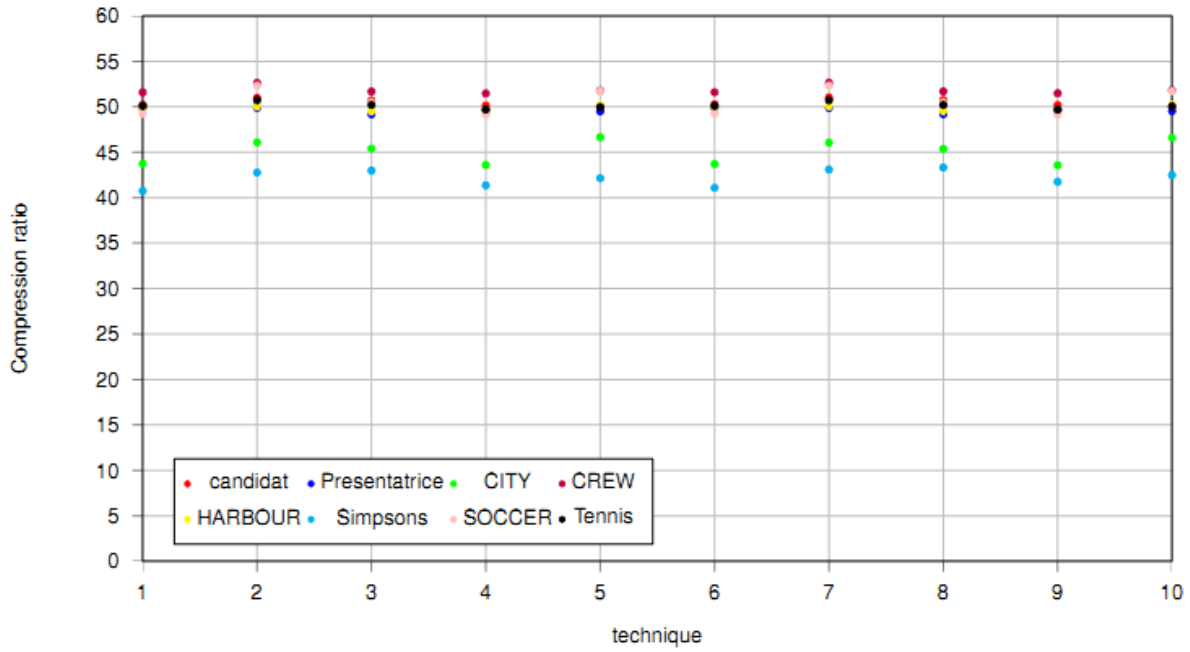


Figure 3- 10 : "*Partial encode system***" compression ratio**

The same conclusion as for the "***close loop adaptation system***" can be drawn as techniques based on random motion vector selection (1 and 6) provide the worse results, while more sophisticated techniques often provide results that depend on the video content.

2.2.4 Performance evaluation of the "generic intra refresh adaptation system"

Finally, the same evaluation has been realized on the "***generic intra refresh adaptation system***". This adaptation system is simpler than the previous ones. The major difference comes from the fact that the "***intra refresh system***" does use motion compensation in the encoding path in order to save computational complexity. As a result, the mismatch between estimated pixels and estimated motion vectors are not compensated any more during the video re-encoding stage. This is what explains quality drops shown in Figure 3- 11.

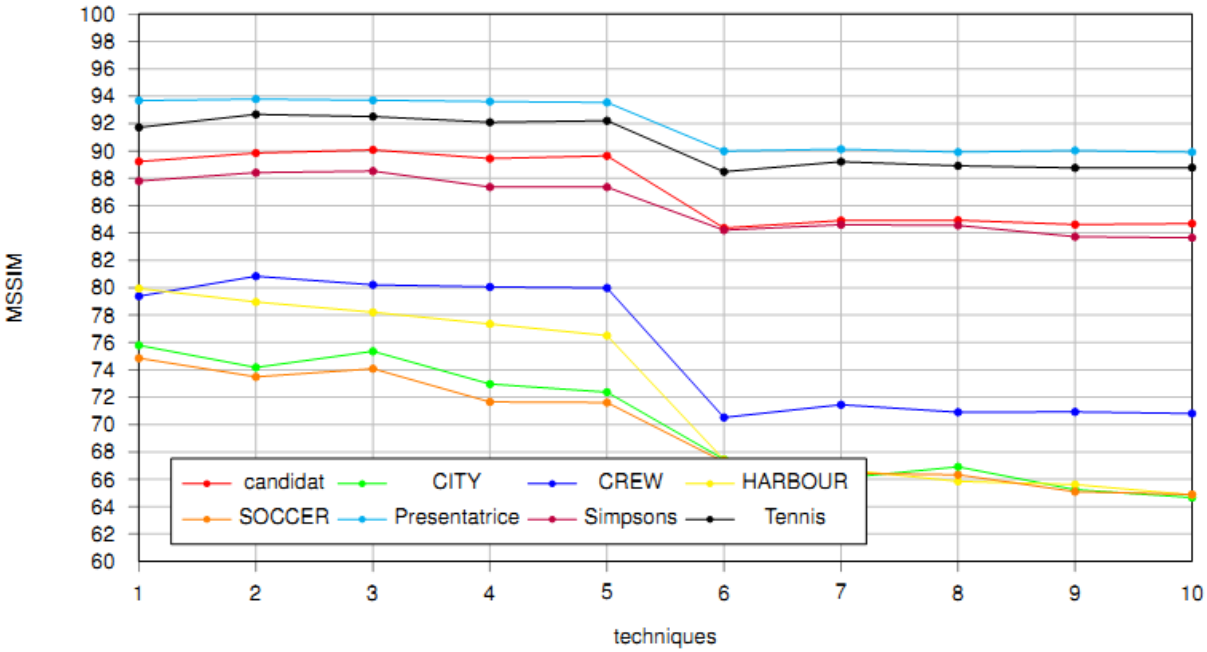


Figure 3- 11 : Result SSIM for "*Generic intra refresh adaptation system*"

Video quality results are lower than the previous adaptation system results because of the lack of estimated error correction. Estimation errors are detrimental because they generate additive noise to the decoded video stream. Indeed, computation errors of frame (i+1) are added to the ones of frame (i), etc. until the video decoder finds an Intra-predicted frame. This explains why some videos having a large set of motion provide low quality scores.

Least impacted video are "Presentatrice", "Tennis" and "Candidat". Those video are characterize by almost no motion ("Presentatrice", "candidat") or few simple motion ("Tennis"). On the contrary, video with a lot of motion are greatly impacted by the adaptation system ("SOCCER", "CITY"). These results illustrate that mismatches between estimated motion vectors and estimated pixels are not corrected during the encoding step. Video with few or no motion vector are well adapted (good resulting quality) as there is few or no mismatch. However, adapting video with a lot of motions will generate a lot of mismatches that will not be compensated resulting in quality loss.

In Figure 3- 12, the compression ratio for each technique and video is displayed. Like for other adaptation systems, the motion vector and pixel merging technique impacts the bitstream size. However, the adaptation system does not compensate the estimation errors - the residue set for inter-frame (P) is still low – so the compression ratio obtained for the adapted video stream increases up to 72% (near to the theoretical limit of 75%).

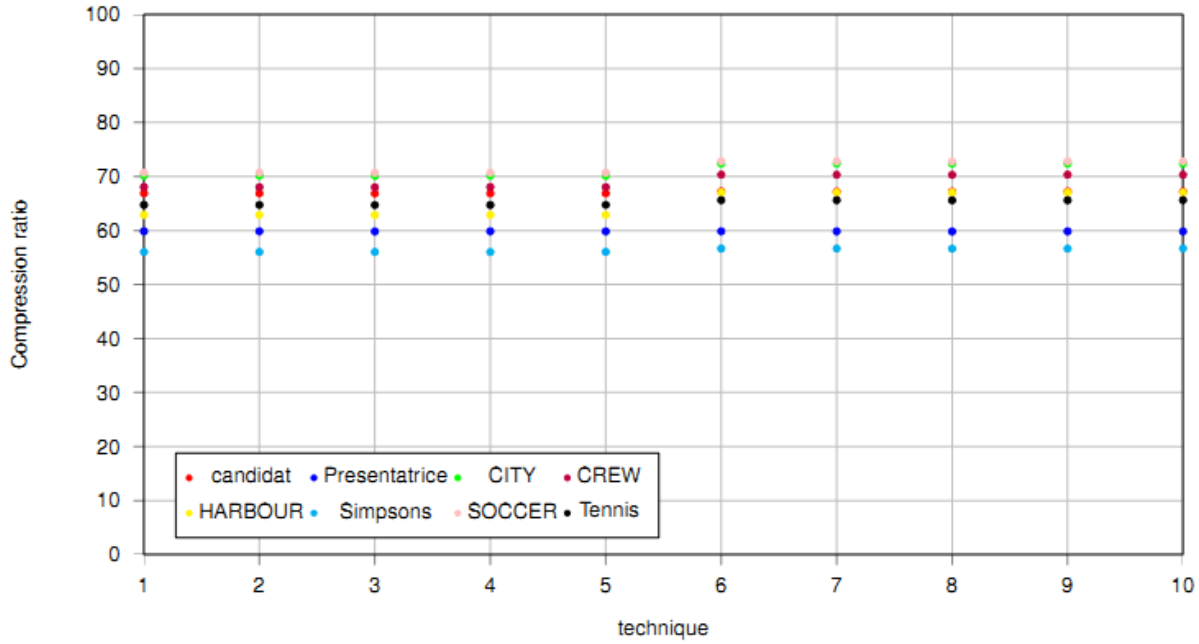


Figure 3- 12 : "Generic intra refresh system" Compression ratio

2.2.5 Video quality during playback

In the previous section, we have compared the adaptation systems based on the average video quality (average SSIM value of frames composing the video). This information is enough to show that the two first systems provide good results, while the last one is the worst one. However, the "***generic intra refresh system***" achieves good results for video with few motions. In order to pursue our analysis, information on the video quality during the adapted video playback has been computed. Table 3- 3 and Table 3- 4 provide the average SSIM values depending on the video set and their associated standard deviation.

The differences between the close loop and partial encode results are very small. Average values are close and standard deviations are very small. The "***intra refresh system***" achieves poor results: average results are low and standard deviations are high. In comparison with the other two adaptation systems, the "***generic intra refresh adaptation system***" is less reliable (high standard deviation) and achieves a low overall quality. Standard deviation results show how much the quality varies over times but not how often a quality over time analysis shall be done.

MEAN	Candidat	CITY	CREW	HARBOUR	Presentatrice	Simpsons	SOCCER	Tennis
Close Loop	95.88	98.73	97.02	99.31	97.06	97.27	98.14	97.63
Partial Encode	95.06	98.24	95.93	98.97	96.46	96.84	97.09	96.77
Intra Refresh	87.45	70.71	76.03	72.96	92.05	86.28	70.09	90.73

Table 3- 3 : Mean SSIM for test videos

STD_DEV	Candidat	CITY	CREW	HARBOUR	Presentatrice	Simpsons	SOCCER	Tennis
Close Loop	0.02	0.02	0.01	0.01	0.05	0.04	0.01	0.02
Partial	0.02	0.02	0.03	0.01	0.05	0.04	0.02	0.02

Encode								
Intra Refresh	2.44	3.81	4.52	5.82	1.8	1.79	3.38	1.68

Table 3- 4 : standard Deviation of SSIM for test videos

The quality of experience is not always a matter of overall video quality but also a matter of quality variation. Once a minimum overall quality is met, a video with a constant quality over time is considered better than a video whose quality continuously drops and rises. Figure 3- 13, Figure 3- 14 and Figure 3- 15show SSIM evolution over time for the 3 adaptation systems.

The temporal results confirm the conclusion we made on average SSIM values. "***Close loop system***" and "***partial encode adaptation systems***" achieve almost the same quality, while the "***intra refresh adaptation system***" has a varying quality due to drift error. Additionally, to achieving a poor quality, the "***intra refresh adaptation system***" cannot maintain a constant quality over time and thus is not good enough to be selected for achieving our goal.

2.2.6 Conclusion

In the downscaling frame resolution process, "***generic partial encode system***" and "***close loop system***" have achieved good quality result. On the contrary, the "***intra refresh adaptation system***" has shown results too poor to be further considered as a good candidate.

"***Partial Encode system***" and "***Close Loop system***" seem to be very resilient to the choice of pixel merging and motion vector estimation techniques. Hence, both the quadratic and linear average can be stated as achieving equal results in the adaptation process. Because of that, computational cost is taken into account in order to determine the best technique to employ. Therefore, in the following resizing evaluations, linear average for both the pixel and the motion vector estimations is implemented.

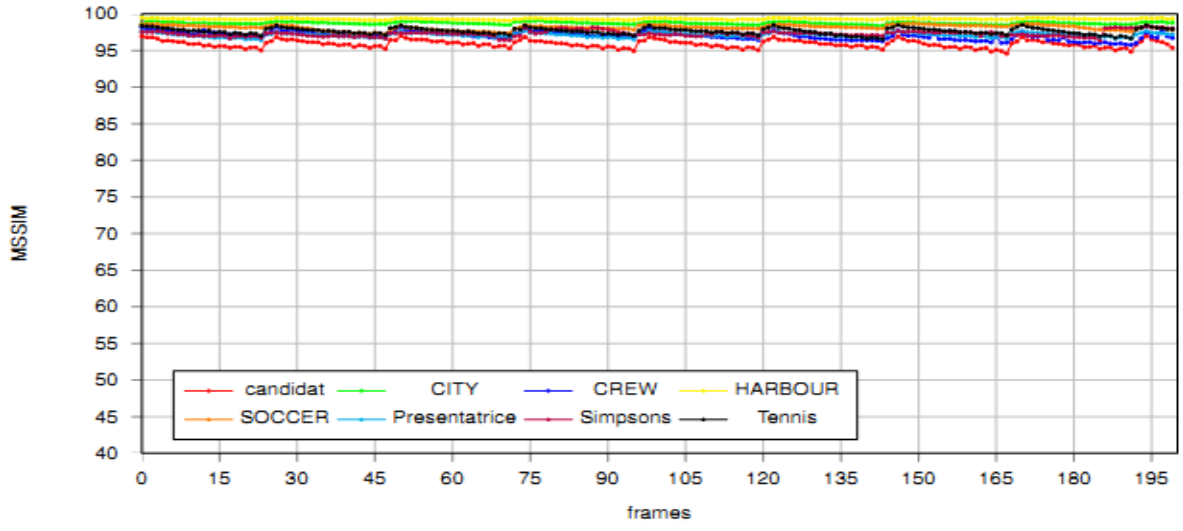


Figure 3- 13 : "Close Loop adaptation system" quality over time

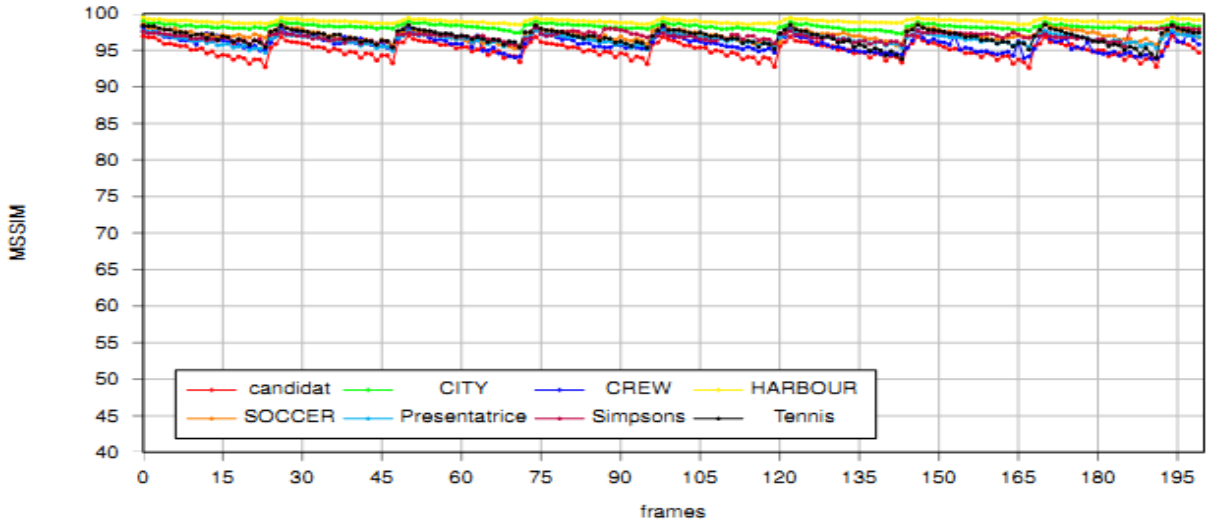


Figure 3- 14 : "Generic Partial encode adaptation system" quality over time

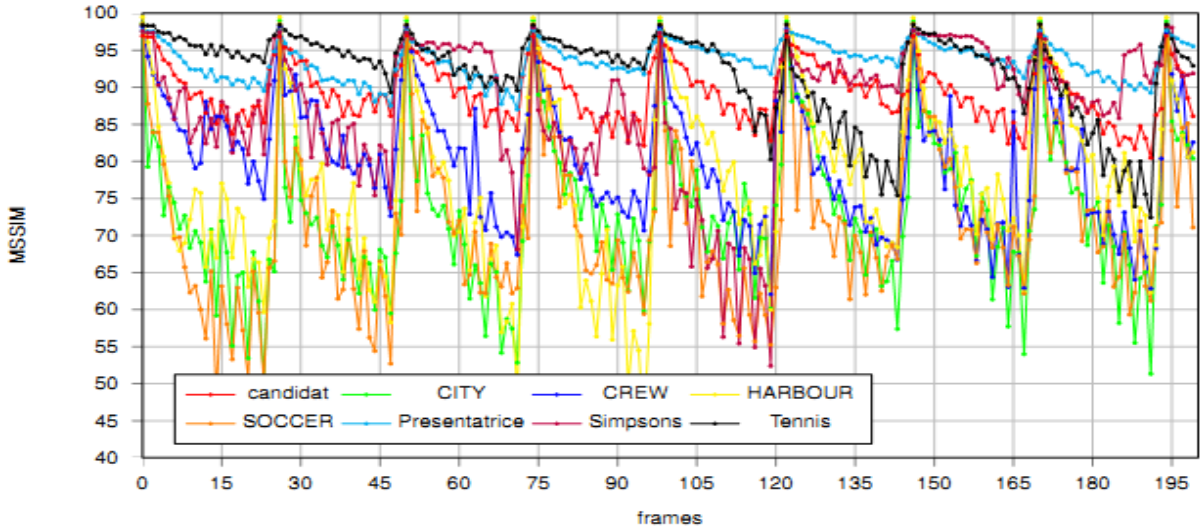


Figure 3- 15 : "Generic intra refresh adaptation system" quality over time

2.3 Bitrate reduction evaluations

2.3.1 Test Bench

The test bench follows the same pattern as for the resolution downscaling test bench:

1. The original video stream is adapted using one of the selected processing chains;
2. The adapted video and the original video are decoded;
3. Both decoded videos are evaluated with the presented quality metrics (section 1.3.3).

We use the same quality evaluation approach than for the downscaling adaptation evaluation. This approach is based on (1) a SystemC model that we have developed and (2) on our QETool. Bitrate adaptation keeps the number of pixels per frame and the number of frames per second unchanged, hence the adapted video can be compared to the original one, without previous computation.

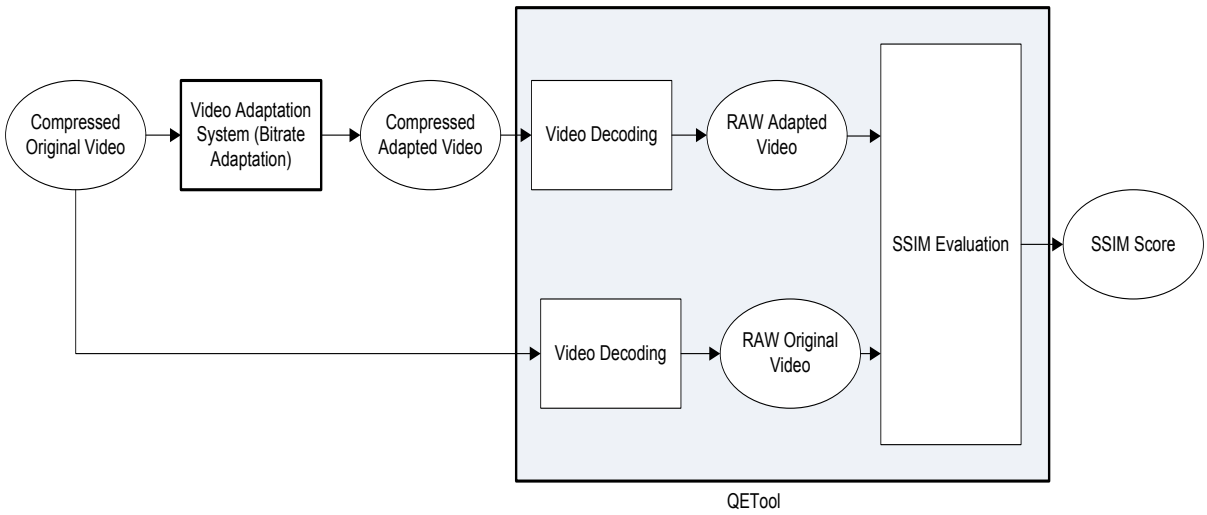


Figure 3- 16 : Bitrate adaptation Evaluation system

The behavior of the two remaining adaptation systems will be obtained by successive bitrate reduction. Our bitrate diminution implementation takes place at the frame level and aims at reducing the bit cost of each frame by the specified reduction amount. The implementations do not take headers into account, such as the GOP header or the global video header. Those headers require bits to be coded which are not taken into account in our algorithm. However, we have taken into account the global video bitstream to measure the overall bitrate diminution. Hence, there is a mismatch between the aimed bitrate reduction and the measured bitrate. Other studies may aim at (1) reduces mismatch between desired bitrate and obtained bitrate and (2) having a flat bitrate evolution over time. For this study, this mismatch is not troublesome as we aim at evaluating rough quality evolution with bitrate reduction.

2.3.2 Global Results

Results for each video have been averaged to compare the overall results of both the "***close loop system***" and the "**generic partial encode adaptation system**". For average to have meaning, standard deviation has also been computed to represent the reliability of the average result. Small standard deviation means high reliability of the average result where high standard deviation means that the average result is not reliable. Figure 3- 17 shows the results with the standard deviation (represented as error bars). The four points represent the four bitrate reduction aims. These four points are 10%, 20%, 30% and 40% bitrate reduction. As stated above, this study is not about precision aims but quality evaluation which explain that bitrate reduction targets are not satisfied. Both adaptation systems achieve good results between 96% and 99% with the "***close loop adaptation system***" slightly above the "***generic partial encode system***".

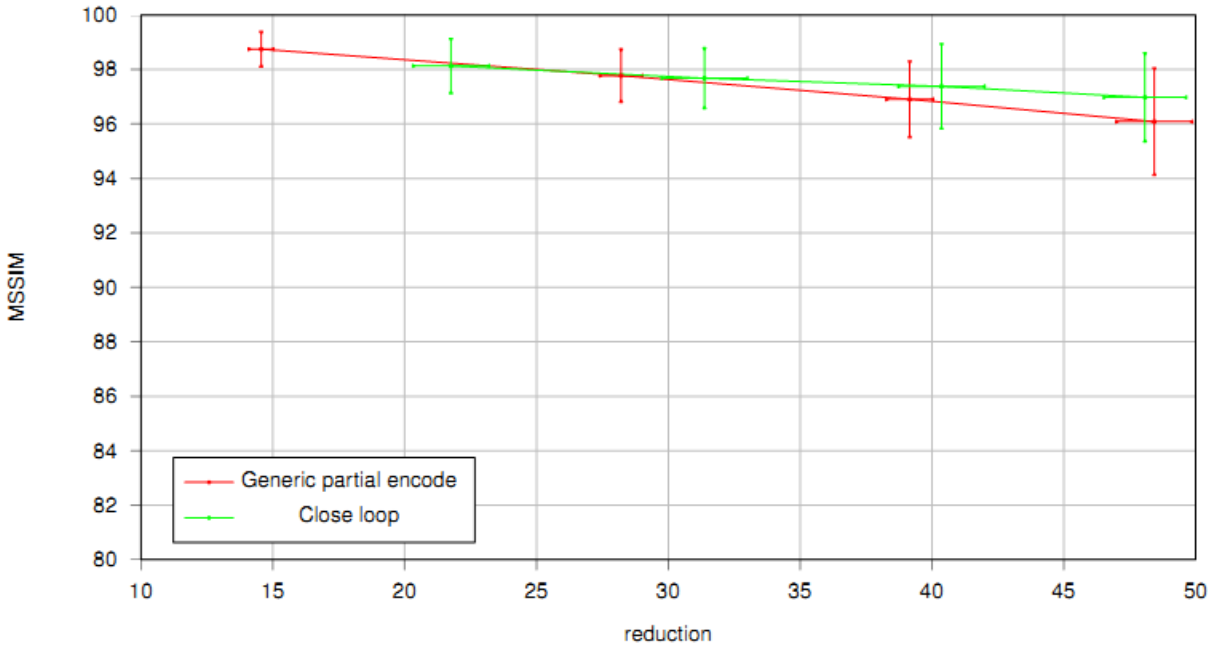


Figure 3- 17 : Bitrate reduction - Adaptation system Comparison

2.3.3 Temporal Results

Figure 3- 17 shows the temporal evolution of the quality for the "candidat" video after a bitrate reduction of 40%. The same behavior is observed for both adaptation systems. Intra (I) frames have a great quality, while INTER (P or B) frames are more impacted by the bitrate reduction operation. This is why both curves seesawed. After an I frame, drift effect occurs on successive P and B frames steadily reducing frame quality until another I frame occurs resetting the errors and restoring the quality.

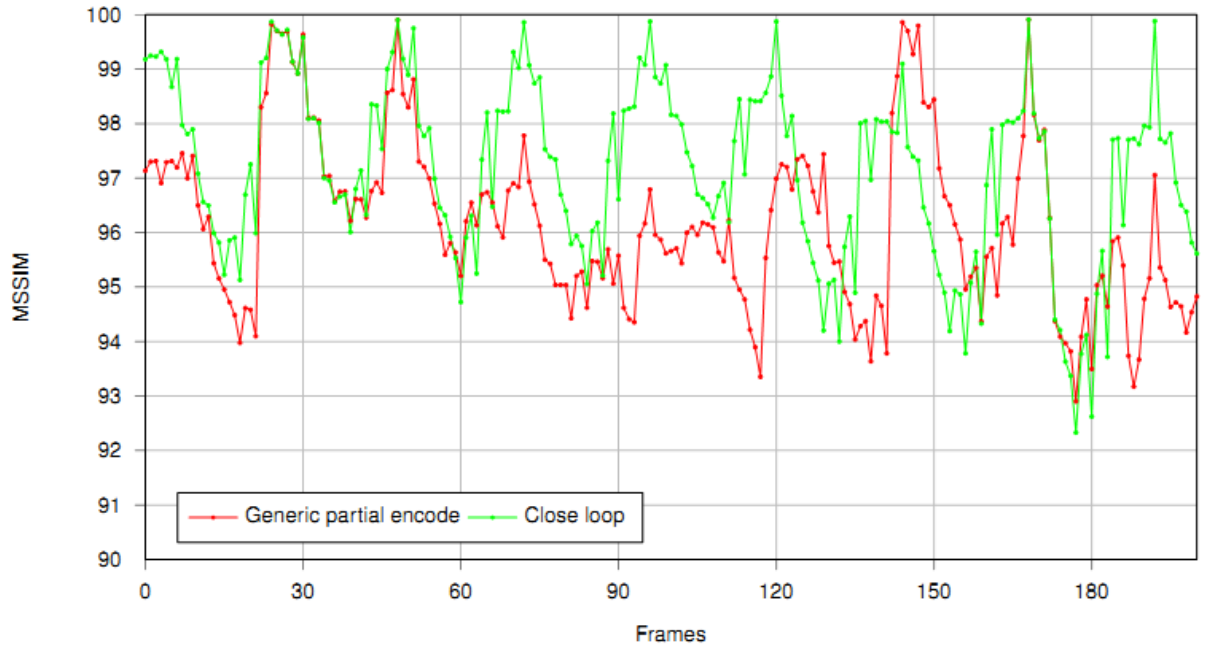


Figure 3- 18 : MSSIM over time for "candidat" video with bitrate reduced of 40%

There is no huge gap of quality over time, the MSSIM score is held between 100% and 92% for the whole time. The "***Close Loop adaptation system***" achieves for almost every frame a better quality than the "***generic partial encode system***". However, quality differences are not so distinct, as the maximum quality difference between two frames is 8%, in the 40% bitrate reduction process. To illustrate such feature, Figure 3- 19 and Figure 3- 20show the same frame processed by the "***Generic partial encode system***" and the "***close loop adaptation system***". These frames correspond to the ones with the highest difference (8%) in quality between the two video adaptation systems. Actually, no differences are noticeable between these two frames which confirms that quality difference between the two adaptation system are slims.

Figure 3- 19 : Close Loop:"SOCCER" 40% bitrate reduction

Figure 3- 20 : Partial Encode: "SOCCER" 40% bitrate reduction

2.3.4 Conclusion

The two remaining adaptation systems have been tested regarding the bitrate adaptation process. While the close loop system achieves the expected best quality result among the two systems, fair quality results are obtained by the generic partial encode system. As a proof, a video adapted by the "***generic partial encode system***" is hardly seen by a human as different as a video adapted by the "***close loop system***".

2.4 Conclusion

In this chapter, first, previously presented adaptation systems have been analyzed in order to select those that can achieve a generic adaptation. From this analysis, the only adaptation system presented in chapter 2 that can be considered generic is the "***close loop system***". However, the analysis has outlined features that a generic adaptation system should have. Thanks to this outline, we have proposed two more generic adaptation systems: (1) the "***generic partial encode***" and (2) the "***generic intra refresh***".

These three adaptation systems have been tested regarding the frame resizing issue and the bitrate adaptation issue. In terms of quality and compression ratio, the "***close loop adaptation system***" achieves the best results of the three adaptation system. The "***generic intra refresh***" achieves poor results for the frame resizing issue due to the lack of reference in the encoding process. However, the "***generic partial encode***" achieves results close enough to those of the "***close loop system***" for not being noticeable by human eyes while being less computational than the "***close loop system***".

This study has been used to define the adaptation system to be used for achieving our objectives presented in Chapter 1. As well, this adaptation system has been selected to be implemented in the ARDMAHN project that shares common objectives with our work as presented in chapter 1. The "***generic partial encode***" has been chosen as it is less computational and produces results similar in observed (subjective) quality as the "***close loop***". It is selected as the generic video adaptation system, integration on FPGA and the associated issues will be tackled in the next chapter.

Chapter 4 : A generic multi-codec FPGA based architecture

In chapter 3 we have evaluated the video quality achieved using existing and proposed adaptation systems. At the end of the chapter, we have proposed a custom adaptation system that enables video resizing and bitrate control independently of the video codecs. Video quality evaluation concluded that the proposed adaptation system achieves a good tradeoff between quality and computational complexity in both use cases.

In this chapter, the codec adaptation issue is first addressed as an architectural constraint. An architectural solution based on partial dynamic reconfiguration available in nowadays FPGA is presented. Then, the system and the adaptation architecture are presented. Finally, the FPGA implementation of the system and the hardware solution for a complete MPEG-2 adaptation system are provided.

1 Video adaptation generic design

In chapter 2, an adaptation system overview has been presented. Literature adaptation systems only focus on a specific adaptation issue such as MPEG-2 frame resizing or MPEG-2 to H.264 transformation. There is no generic adaptation system that solves, in a generic way, any kind of adaptation (any codec, any adaptation).

Hence, to implement a multi-standard, multi adaptation system, one has to implement every single combination of encoder, decoder and adaptation. Figure 4- 1 (a) illustrates this issue. This means that for N_S video codecs there are N_S video coders and N_S video decoders, hence N_S^2 configurations to be implemented. With N_A adaptations, there is a need to implement N_s^2 x N_A dedicated adaptation systems. Developing this number of configurations is time consuming and requires large – and costly - devices to implement the system in parallel.

To overcome this issue, we can reuse already implemented designs (Figure 4- 1 (b)). Reusability reduces the codec implementation cost to 2 x N_S instead of N_S^2, thus drastically reducing time to market and development costs. To adapt the encoder and the decoder according to the adaptation requirements, partial dynamic reconfiguration can be used to reduce hardware area. This area saving is obtained by modifying FPGA configuration in loading/removing components depending on the adaptation context so that every adaptation system is only loaded in the device when needed.

Reusing codec implementation helps reducing implementation costs. Adaptation implementation can also be reused. Indeed, video adaptation categorization (frame resizing, frame skipping, bitrate reduction) does not depend on video standard. Hence, adaptation operates the same final transformations (reducing the frame resolution or frame rate or video bitrate) but algorithms may be different because they operate on different parameters (motion vectors, quantizer scales …) defined by the standard. Hence to reuse adaptation implementation, it is mandatory to have a generic adaptation implementation. This generic adaptation must operate on a unique set of parameters. We will refer to this unique set of parameters as the unified format. This unified format will be detailed after discussing its advantage.

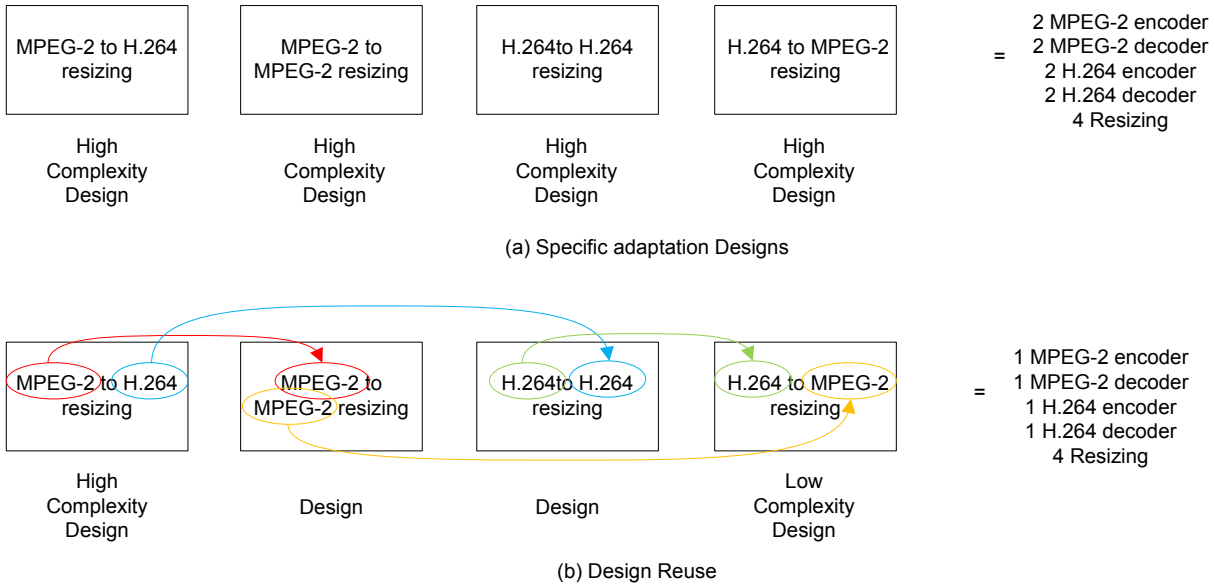


Figure 4- 1 : Design Reuse Example

Thanks to the unified format that fulfills the video codec requirements, the generic video adaptation design may be performed by design composition as depicted in Figure 4- 2. A specific adaptation is done by selecting the decoding/encoding/adaptation path from a design pool. Those paths are seamlessly communicating in the same unified format. Thus, only three generic adaptation implementations are needed: (1) frame resizing, (2) frame skipping and (3) bitrate reduction. Implementation cost of a fully generic adaptation system is reduced to 2 x N_S+3.

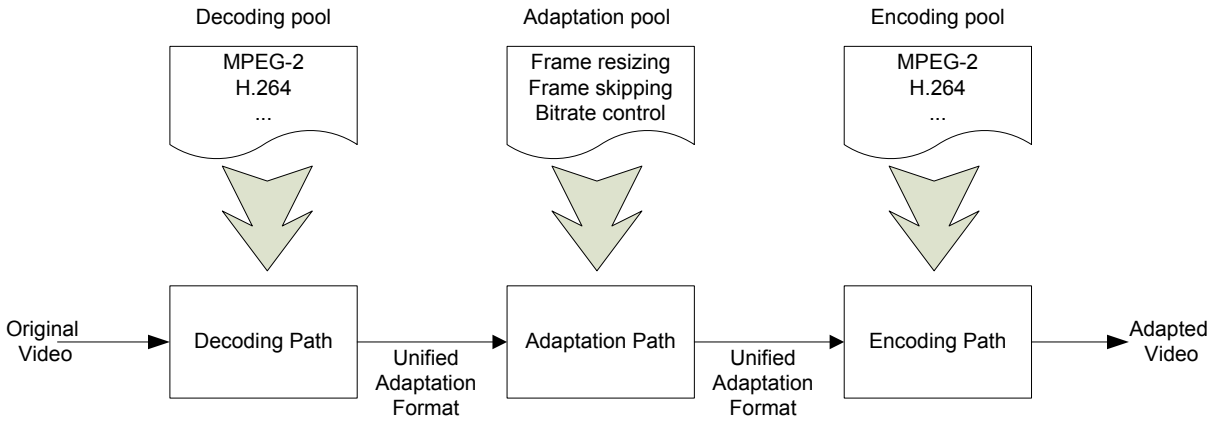


Figure 4- 2 : Generic Adaptation Framework

The next two sections, we will focus on what is required for the adaptation module taking place in the adaptation path to be generic before defining the unified format composition.

1.1 Adaptation implementation considerations

In the adaptation path, three adaptations may take place: “frame skipping”, “frame resizing” and “bitrate control”. An adaptation scenario may need from 0 up to 3 adaptations on the same video - e.g. ”no adaptation” or “frame skipping” + “bitrate control” or “frame skipping” + “frame resizing” + “bitrate control”. Any adaptation combination can be required. To implement such system, different solutions exist.

A first solution would be to create as many designs as there are adaptation possibilities. With 3 developed adaptations, there are 2^3 = 8 possible designs. Development time is the same but there are 8 times more designs than having a unique adaptation.

A second solution is to develop a unique adaptation design that is able to perform any adaptation combination. To select the proper adaptation combination a “selection input signal” shall be used. This unique adaptation can be designed considering two concepts:

- First and foremost, the adaptation implementation can be generic enough to process the identity due to specified characteristics. For example, a frame resizing that takes the resizing ratio to operate. If the specified resizing ratio is 1, the process resizes the frame by a ratio of one, which means that the output frame size is the same as the input frame size. Hence, the adaptation always operates but it could have no effect.
- Second, a “trigger signal” can be used to enable the proper adaptation combination. This signal can be used in addition to clock gating techniques (Figure 4- 3) that power down unused FPGA region to reduce power consumption of the global system.

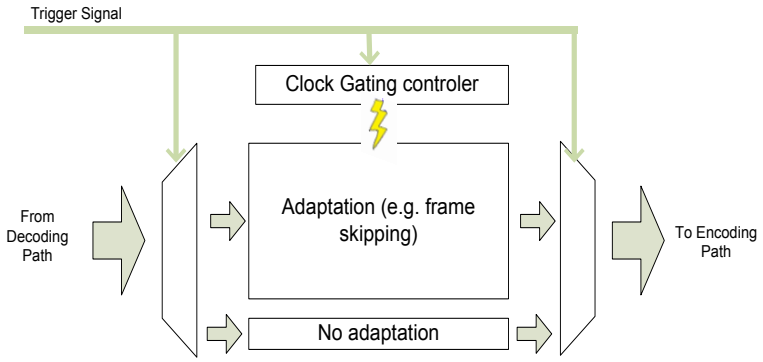


Figure 4- 3 : trigger signal and clock gating

Video adaptation can be processed in any order. However, an optimization can be reached by selecting the right order. Bitrate adaptation techniques are based on removing information to reduce the number of bit needed to encode a frame. The number of information to keep or remove is based on a “bit budget” allocated to each frame according on various parameters such as the desired bitrate, the number of frame per second, the type of the frame … This “bit budget” is linked to the number of pixel in a frame (frame resolution) and the number of frame per second (fps). For a given bitrate, the less fps, the more “bit budget” there is for other frames in the GOP. Similarly, the fewer pixels there are in a frame, the more “bit budget” there is for each macroblock in the frame. Hence, video bitrate adaptation process shall be the last adaptation process when triggered.

In order to avoid useless computation, frame skipping process shall be the first adaptation process (if triggered). Indeed, if the frame skipping process operates after the frame resizing process, the frame resizing process will operate on frames that will be removed afterwards by the frame skipping process. Thus, to optimize the adaptation process by avoiding useless operation, the frame skipping process shall be the first video adaptation process among the three, leaving the second place to the frame resizing process. This order is depicted in Figure 4- 4.

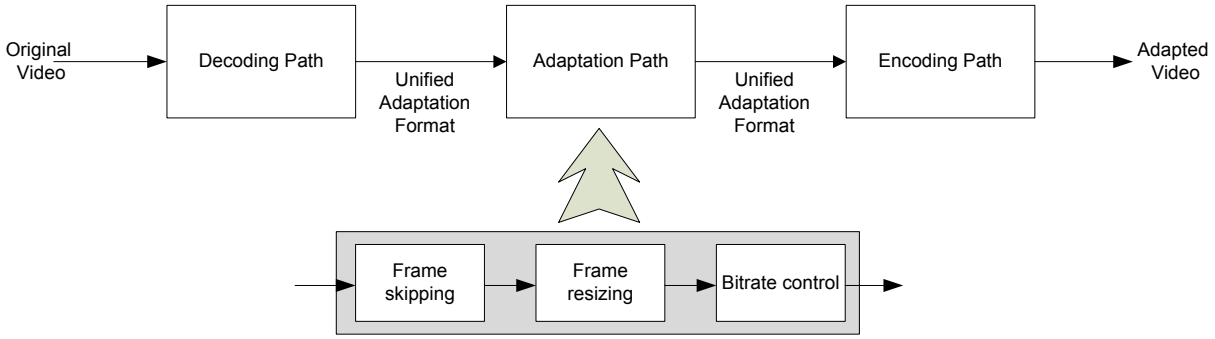


Figure 4- 4 : Video adaptation process in optimized order

1.2 Unified adaptation format

Manipulating design reuse to optimize development costs and system resources requires developing under a specified framework. As mentioned above, this framework implies a unique data format to seamlessly interconnect any decoder/encoder designs to the adaptation process. This format shall be precise enough so that the maximum video standard features can be used. For example, the H.264 standard uses ¼ pixel precision motion vectors but the MPEG-2 standard only uses ½ pixel precision motion vector. It is then mandatory that the unified adaptation standard specifies ¼ pixel precision motion vector so that both the MPEG-2 and the H.264 standard can be fully implemented. Hence, a standard feature comparison will be held before defining the unified adaptation format.

Table 4- 1 sums up the different standards and their associated features in used in nowadays' multimedia content. Every standard supports Intra, Predictive and Bidirectional picture coding (except H.261 which only handles I and P types). The most precise motion vector resolution is ¼ pixel precision for MPEG-4, H.264 and WMV9/VC-1 standards. The most precise transform operates on 4x4 blocks (WMV9/VC-1 and H.264), which is the exact same size as the smallest vector block size (H.264). Only H.264 supports spatial intra prediction. Field and frame prediction mode as well as progressive and interlaced formats are supported by a major part of the existing standards. The entropy coding feature and the presence of a de-blocking filter are not useful for the adaptation process.

Features	H.261	MPEG-1	MPEG-2	H.263	MPEG-4	H.264	WMV9/VC-1
Picture Coding Type	I, P	I, P, B	I, P, B	I, P, B	I, P, B	I, P, B	I, P, B
Entropy Coding	VLC	VLC	VLC	VLC, SAC	VLC	UVLC, CAVLC, CABAC	Multiple table VLC
MV Resolution	Int. Pel	½ pel	½ pel	½ pel	¼ pel	¼ pel	¼ pel
Transform	8×8 DCT	8×8 DCT	8×8 DCT	8×8 DCT	8×8 DCT	4×4 & 8×8 Integer	8×8, 8×4, 4×8, 4×4 Integer DCT
Vector Block Size	16×16	16×16	16×16, 16×8	16×16, 8×8	16×16, 8×8	16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4	16×16, 8×8
Spatial Intra Prediction	No	No	No	No	No	Yes	No
Formats Supported	Prog.	Prog.	Prog./Intr.	Prog.	Prog./Intr.	Prog/Intr	Prog/Intr
Prediction Modes	Frame	Frame	Field & Frame	Frame	Field & Frame	Field & Frame	Field & Frame
De-Blocking Filter	In-loop	None	Post	Annex J In-loop	Post	In-loop	In-loop

Table 4- 1 : Standard feature summary

Nowadays, the standard that most gathers feature is the H.264 standard. Hence, to support as many standards as possible, H.264 features are used as models to the unified adaptation format definition. Macroblocks represent 16x16 regions of the frame in the YUV color space (see chapter 2). It is then possible to receive data in 16x16 blocks with the corresponding metadata specified above. However, the smallest vector block size is 4x4 and there are disparities between standards. One macroblock

may possess up to 32 motion vectors but can change from one macroblock to another. Most of the adaptation algorithms (especially frame skipping and frame resizing) assume that one macroblock has up to 2 motion vectors.

To solve this issue, there exist two possibilities:

- A new adaptation algorithm that supports up to 32 motion vectors per macroblock shall be found
- Another representation allowing up to 2 motion vectors shall be used. Thus, existing algorithms can be used on this representation.

We have investigated the second proposition to find a representation that will enable the use of already proposed adaptation algorithms. For this reason, the unified adaptation format embeds 4x4 pixels along with its corresponding macroblock metadata and thus its corresponding motion vectors (up to two). Hence, macroblocks pixels have to be separated in 4x4 blocks and macroblock metadata have to be duplicated and mapped to the aforementioned 4x4 block. To reform a "standard" macroblock, 16 4x4 blocks are merged into a unique 16x16 blocks, the 16 metadata structures are mapped back to a unique metadata structure that comes along with the 16x16 block. Two macroblock translation proposals for MPEG-2 and H.264 are detailed in the following sections.

1.2.1 MPEG-2 format to/from unified representation format

MPEG-2 uses from 0 (I frame) up to 2 (B frame) motion vector per macroblock, representing 16x16 pixel in the coded frame. Thus, developing a 16x16 macroblock into small 4x4 macroblocks implies full duplication of the macroblock metadata as illustrated by Figure 4- 5.

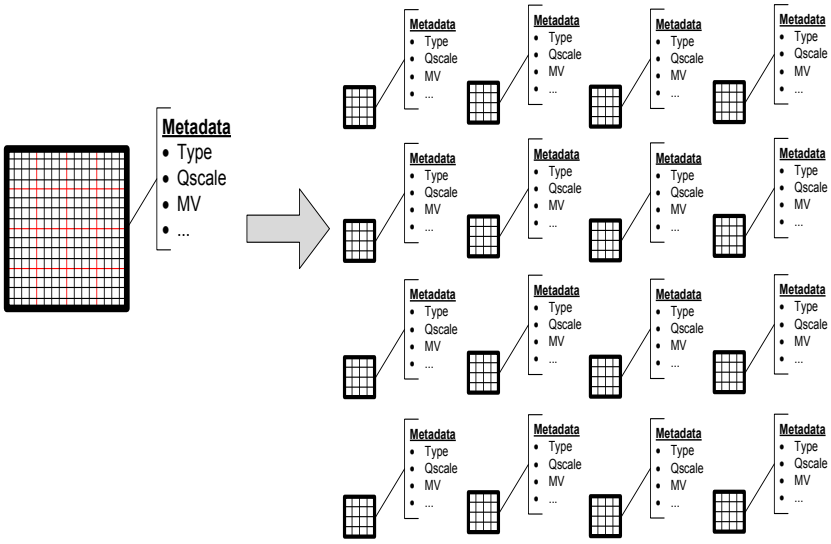


Figure 4- 5 : Macroblock division and metadata duplication

To merge back the 16 blocks, pixel data and metadata shall be merged. To merge pixel data, only data reordering is used. However, to merge metadata, reordering is not enough: (1) the adaptation process may have changed some metadata or (2) metadata comes from other standards that have a finer macroblock description (e.g. h.264).

To merge the 16 "macroblocks" into a single one, we propose to use resizing techniques, as described in chapter 2. For example, averaging motion vector, taking the minimum quantizer scale

and giving priorities to Intra macroblocks over Inter macroblocks are the techniques employed in the ARDMAHN project to form the resulting MPEG-2 macroblock.

1.2.2 H.264 format to/from unified adaptation format translation

To translate an H.264 macroblock into several "adaptation" macroblocks, the same duplication technique as described for MPEG-2 is used. However, H.264 has a particular motion vector definition with variable block size. If a 4x4 block has its own motion vector, the motion vector is directly mapped to the associated metadata. If a motion vector designs a larger block, it is then duplicated to the 4x4 blocks composing the larger region. This process is illustrated by Figure 4- 6.

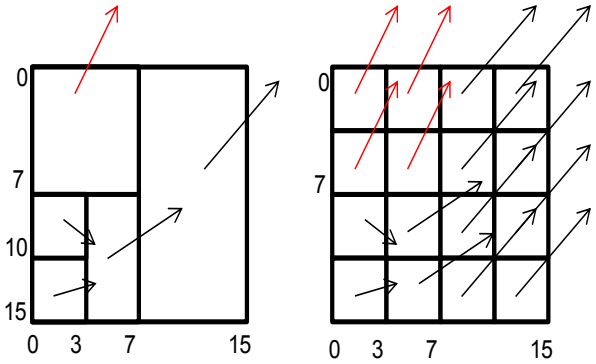


Figure 4- 6 : Vector mapping for h.264 to adaptation format translation

To recreate the proper motion vector mapping, we propose to consider a two-step recombination. The first step operates on a 2x2 array of 4x4 blocks. Every 4 motion vectors are compared and, upon equality, blocks are merged according to Figure 4- 7.

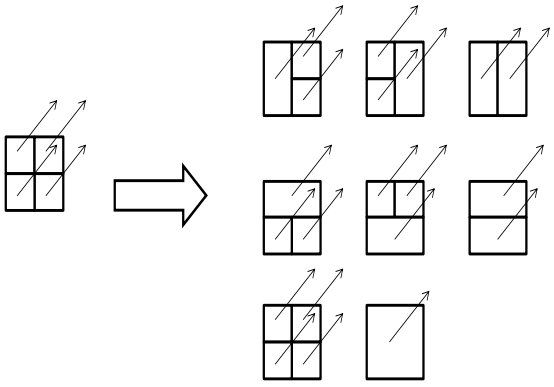


Figure 4- 7 : H.264 motion mapping

This mapping is done for the four 2x2 array composing the 16x16 macroblocks. Every array may result in one of the eight merged blocks. This first mapping is done again in the second step. However, only blocks containing one motion vector are selected for this second mapping. This two-step reconstruction is illustrated by an example in Figure 4- 8 (first step) and Figure 4- 9 (second step).

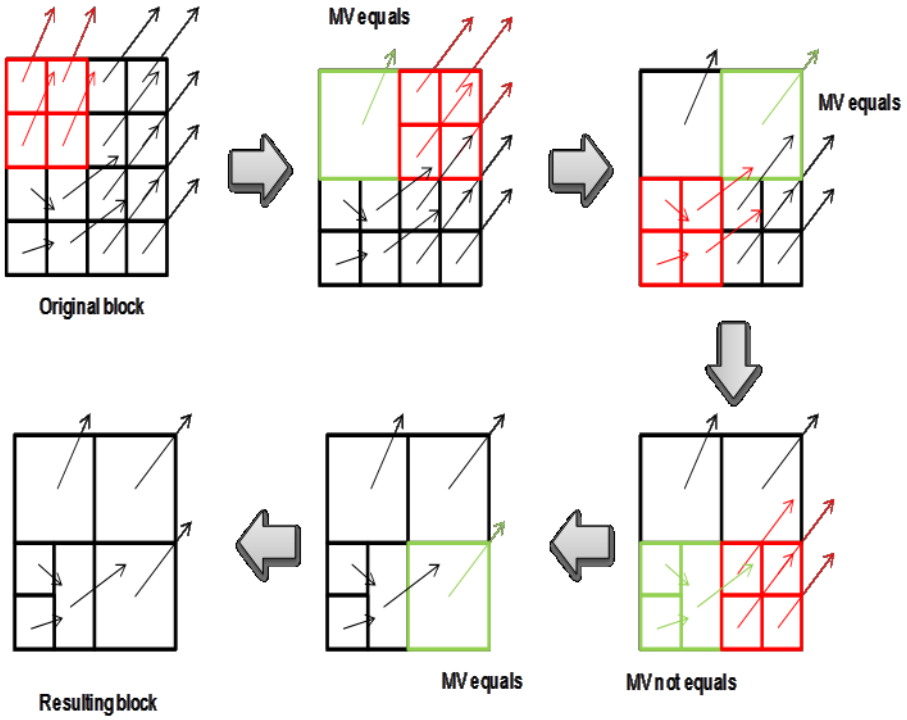


Figure 4- 8 : First step recombination

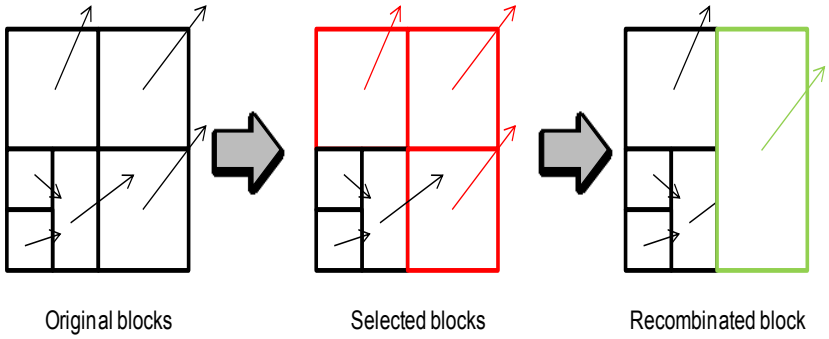


Figure 4- 9 : Second step recombination

1.3 Conclusion

Designing hardware task is very time consuming. In order to reduce development cost, our proposal is to re-use already developed designs. For video adaptation, this means not only re-using already developed encoding and decoding paths but also having a unique adaptation path. Hence, every kind of specific adaptation design can be created by combining a specific set of codecs along with a generic adaptation path. This can be achieved by using a unified data exchange format.

The previous section described how to design a generic adaptation path. The proposed unified adaptation format has been presented along with some examples on how to interconnect video codecs with this generic adaptation.

In the next section, a FPGA implementation is considered. Static and dynamic reconfigurations are tackled. FPGA architectures are proposed that enable design reuse thanks to FPGA features.

2 FPGA implementation of a generic adaption system

2.1 Structuration and static reconfiguration

Field Programmable Gate Arrays (FPGA) are programmable circuits consisting of a (1) logic cell network with (2) input-output cells and (3) flexible interconnections resources [XIL00] (Figure 4- 10). Most of FPGA possess more resources than just the aforementioned three. These other resources - DSP, memory … - are used to optimize FPGA effectiveness. Their flexibility associated to their high performances (achieved in case of parallel computation) makes them attractive for high-complexity application under real time constraint [VUI96].Configuring an FPGA means: (1) configuring the logic cells to perform the desired logic, (2) configuring the interconnection between logic cells and (3) configuring interconnection between logic cells and input-output cells.

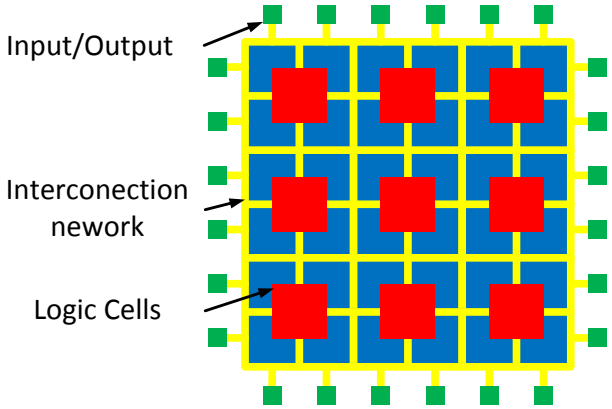


Figure 4- 10 : Basic FPGA Architecture

In order to configure a FPGA, one has to use a configuration bitstream. This bitstream is a binary file with configuration information. The bitstream configures the whole FPGA resources at once in a so called static configuration. Until recently, every FPGA designs use static configuration. The FPGA is configured once to process a single function. This technique is used in hardware accelerator conception for either prototyping or small market sells. Designing circuit for FPGA is a complex and time consuming task compared to software conception.

The static bitstream always describes the entire FPGA chip. Hence, only one design can be processed at a time. This constraint limits the flexibility of the FPGA use. Figure 4- 11 illustrates such constraint. Let us define two bitstreams B_1 and B_2 that configure the FPGA to process two different applications. Designs corresponding to such applications are named D_1 and D_2. Using D_1 and D_2 with B_1 and B_2 at the same time is impossible because either B1 or B2 will be loaded as the last loaded design will erase the first loaded design. The only solution consists in developing a third bitstream B_3 describing the whole chip with D_1 and D_2 together.

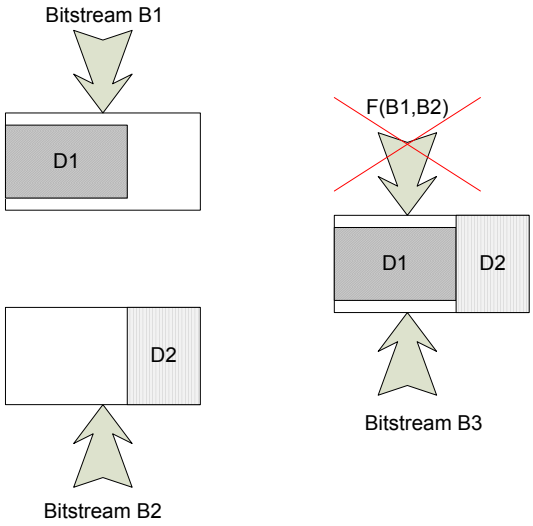


Figure 4- 11 : FPGA static configuration

We can conclude that depending on the number of applications that can run at the same time on the FPGA device, the number of systems to describe grows exponentially limiting the practical use static configuration as the number of bitstreams will be too large.

2.2 Partial dynamic reconfiguration

ATMEL and XILINX have teamed up to design FPGA architectures with on the fly reconfigurable capabilities allowing the so-called dynamic reconfiguration. The Dynamic reconfiguration technique enables the system to configure part of itself during runtime. This adaptation is limited by the time required to reconfigure its design (from millisecond to nanoseconds) [LIU08]. Unlike static reconfiguration, dynamic reconfiguration has the capability to reconfigure the whole or part of the system as many times as necessary and without having to restart it.

Dynamic reconfiguration uses bitstreams that configure only a part of the FPGA. Thanks to such partial bitstreams, designers and researchers have seen the possibilities to create new kinds of FPGA designs with automatic adaptation features. These features require on the fly reconfiguration capability of the logic circuit. Low level FPGA do not allow such features and requires stopping the system in order to load a new configuration.

Designing dynamic reconfigurable systems is more complex than designing static ones. Static systems are described using hardware description language that is translated to target FPGA resources description. These resources are placed and routed considering FPGA resource spatial composition to create the global bistream. This process is well mastered and can be automated.

The dynamic reconfiguration technique requires a more complex design process. Even if some parts are being dynamically reconfigured, the FPGA keeps being defined as a whole by a static bitstream (Figure 4- 12). The static bitstream defines the region where partial bitstreams will be loaded dynamically. These dynamically reconfigurable regions (DRR) have to be defined, during circuit design, by their size, form and position inside the FPGA chip. These characteristics cannot change afterwards.

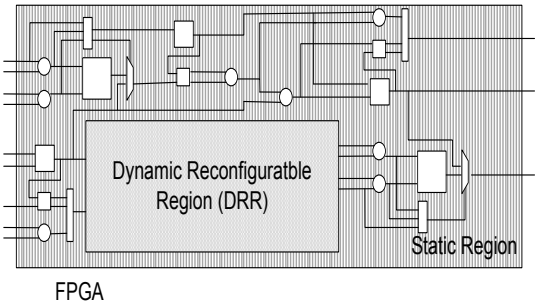


Figure 4- 12 : Partial Dynamic Reconfiguration

Interfacing the DRR with other regions of the FPGA shall be strongly constrained, so that any design configured in the DRR can communicate with the other designs in the FPGA. Interconnections with static regions are by definition static, so any designs that are targeted to DRR shall fulfill these static interconnection constraints.

The dimension and location of the DRR cannot be changed. Hence, the partial bitstream, that will configure this region, configures the whole region as a global bistream configures the whole FPGA. Unused resources by the design cannot be used by other bitstreams, while the design is running on the DRR.

Using dynamic reconfiguration requires possessing various bitstreams that will be loaded in the DRR. This implies that the developed system has at least a memory to store all the partial bitstreams required by the application.

Partial dynamic reconfiguration has opened a new area in the design of FPGA based architecture. Using dynamic reconfiguration enable the use of the time dimension in data processing.

3 Temporal and spatial partitioning

Partial dynamic configurations have led to new ways of designing hardware system on capable FPGA devices. Hardware tasks can be managed by partitioning applications [PUR99] and [CAR03]. Resources allocation is done considering spatial and temporal constraints of the reconfigurable circuit design. When dealing with multiple applications, there exist two forms of partitioning technique:

- Temporal/software partitioning targets application algorithms. It divides every algorithm in temporal execution steps. Multiple applications are handled by switching application at each temporal step (Figure 4- 13).

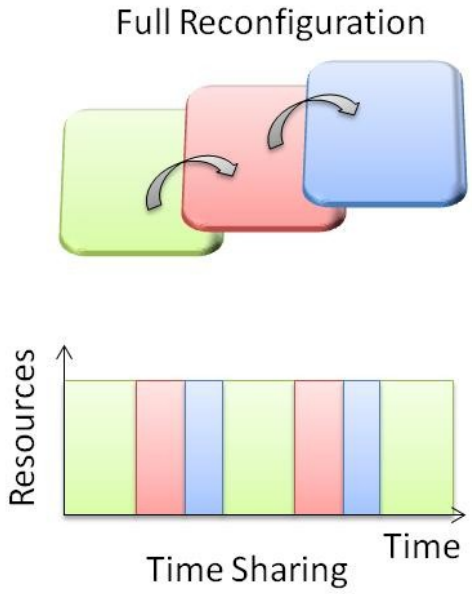


Figure 4- 13 : Temporal Partitioning

- Spatial/Hardware partitioning targets FPGA resources. It reserves FPGA resources for each application. Multiple applications are handled by dividing the FPGA resources and allocating them to each application. Applications are executed at the same time. (Figure 4- 14)

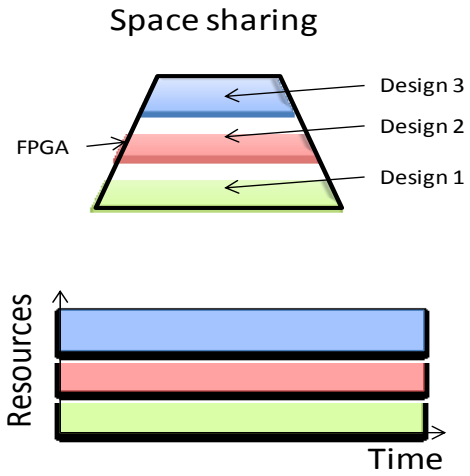


Figure 4- 14 : Spatial Partitioning

Spatial partitioning, presented in Figure 4- 14, authorizes the execution in parallel of the 3 applications, while in case of Figure 4- 13, only one application can be executed at a time. However, while in case of the Figure 4- 14, the system area is equal to the sum of the three application size, the area size presented in the Figure 4- 13 corresponds to the size of the bigger requirements among the three designs.

4 The generic video adaptation architecture

4.1 Static configuration and design reuse

As mentioned above, a certain number of designs have to be created for supporting every kind of adaptation and design reuse greatly reduces this number. The Static bitstreams configure the entire FPGA. Hence, to design a generic video adaptation chip, every adaptation needs to be made possible within the FPGA chip.

A first straight forward design is to elaborate a bitstream for every possible adaptation. Once the encoder, decoder and adaptation design have been created, a “design composition” step has to be performed. This step consists in combining every Decoder-Adaptation-Encoder triplet to form any adaptation design possible and to transform these final designs in bitstream. Toward this, design reuse lessens the development time but does not impact implementation costs that stay very high. Using static reconfiguration requires rebooting the FPGA between two configurations which takes a lot of time. Furthermore, designing a board with an FPGA that should reboot while the rest of the board (external communications for example) is still working can be a difficult task.

To overcome this, we propose a bus-based architecture depicted in Figure 4- 15. The adaptation process is linked by two buses. The input bus links the adaptation process input with the output of video decoding processes. The output bus links the adaptation process output with inputs of video encoding processes. The transfers are controlled by a micro controller.

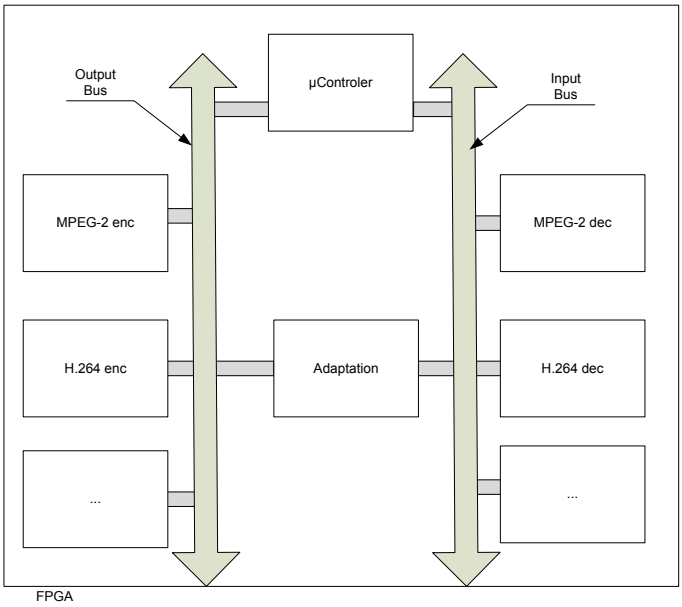


Figure 4- 15 : Bus based static generic video adaptation design

This solution does not require a reconfiguration and thus does not need special powering technique on the board. However, it specifically involves that every codec is implemented on the FPGA at the same time. This implies a lot of FPGA resources and thus an expensive FPGA.

4.2 Dynamic configuration and design reuse

Dynamic reconfiguration is well suited to support design reuse. By allocating each design type (encoder, decoder, adaptation) to a DRR that is configured with the proper bitstream on demand, the design reuse framework is optimally used. Such framework is depicted in Figure 4- 16. This

generic adaptation framework is highly flexible and allows easy codec switching. Decoding, encoding and adaptation paths are configured using available design in its corresponding partial bitstreams pool.

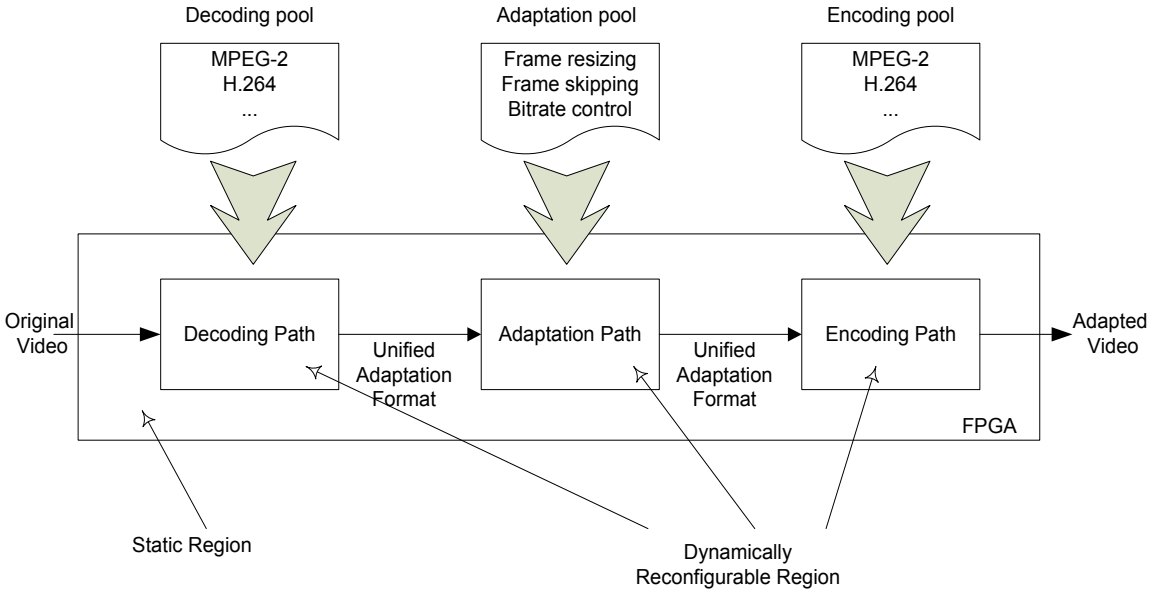


Figure 4- 16 : Generic video adaptation framework

This architecture does not need to have a generic adaptation because it is loaded on the fly. Some situations require no adaptation at all, some others more than one. To fulfill these requirements, the number of adaptation paths can be set from 0 (no adaptation - e.g. only codec change) to 3 (every adaptation at once). Figure 4- 17 and Figure 4- 18 illustrate two examples. Figure 4- 17 illustrates an MPEG-2 to H.264 transcoding with no other adaptation needed. Figure 4- 18 illustrates a frame skipping and bitrate control for a homogeneous MPEG-2 video adaptation.

The possibility of having a varying number of adaptations in the processing chain hits the limitation of hardware dynamic reconfiguration. Indeed, the number of reconfigurable regions and their size are defined at the conception level. Hence, three DRRs are required for the adaptation process. There are two ways of using these three DRRs:

(1) Having a static design that connects the DRR when in use, avoiding the DRR otherwise;
(2) Having a partial bitstream that bypasses the DRR when no adaptation is required.

The second solution is an upgrade of the first one that lowers the resources consumption in the static region. However, if an unique adaptation algorithm is used for each DRR – i.e. one algorithm for frame skipping, one algorithm for frame resizing and one algorithm for bitrate control – then, a static implementation (as described above) should be considered using static region; it is less complex and more easily to optimize than using dynamic region.

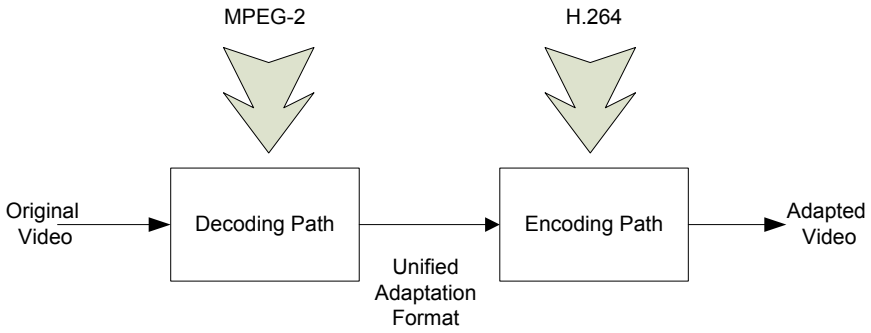


Figure 4- 17 : System implementing a codec adaptation (MPEG-2 to h.264)

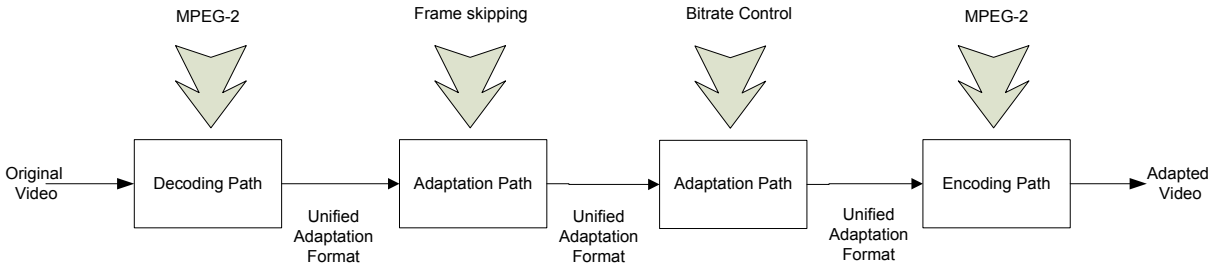


Figure 4- 18 : System implementing frame skipping and bitrate control in MPEG-2 streams

5 ARDMAHN adaptation system

The ARDMAHN project aims at using dynamic reconfiguration with time partitioning to operate multiple video adaptations at the same time with limited resource consumption. The video adaptation system architecture is depicted on Figure 4- 19. Original and adapted video streams are received and sent through PCIe connection to a low cost processor. This processor is responsible for network connection and video pre/post processing (such as header parsing/writing).

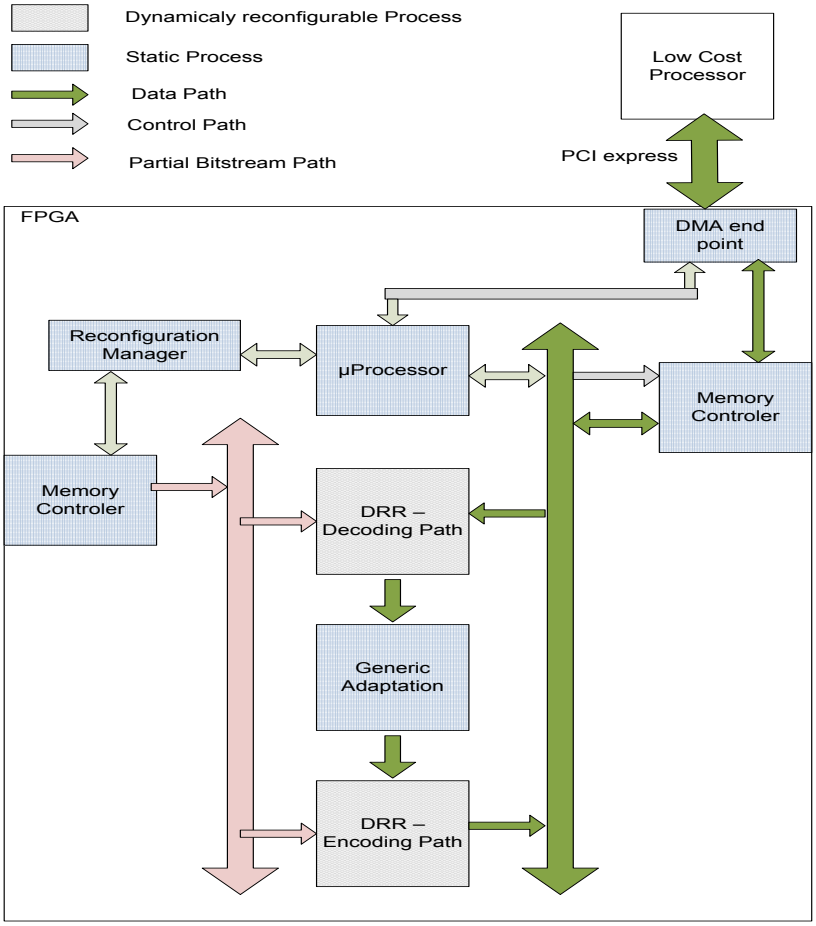


Figure 4- 19 : ARDMAHN Adaptation System Architecture

For the core adaptation system located in the FPGA, two dynamic reconfigurable regions are defined to respectively receive decoding and encoding paths. Those two regions are connected to a static generic adaptation path. Conclusions drawn in chapter 3 considering frame resizing algorithms have shown that our video adaptation system is resilient to the resizing algorithm choice. As a first approximation, this conclusion has been extended from MPEG-2 standard to H.264 standard following the adaptation system depicted in chapter 3. Hence only one implementation of each adaptation process needs to be designed in the static region.

An example of the system evolution over time is depicted in Figure 4- 20. This example shows two different adaptations:

- MPEG-2 to H.264 transcoding with frame resizing and bitrate control adaptation;
- Frame skipping and bitrate control for a homogeneous MPEG-2 video adaptation.

In this example, the decoding path does not change, but the encoding path switches over time from MPEG-2 to H.264 and the adaptation process involves bitrate control but has to switch from frame resizing to frame skipping.

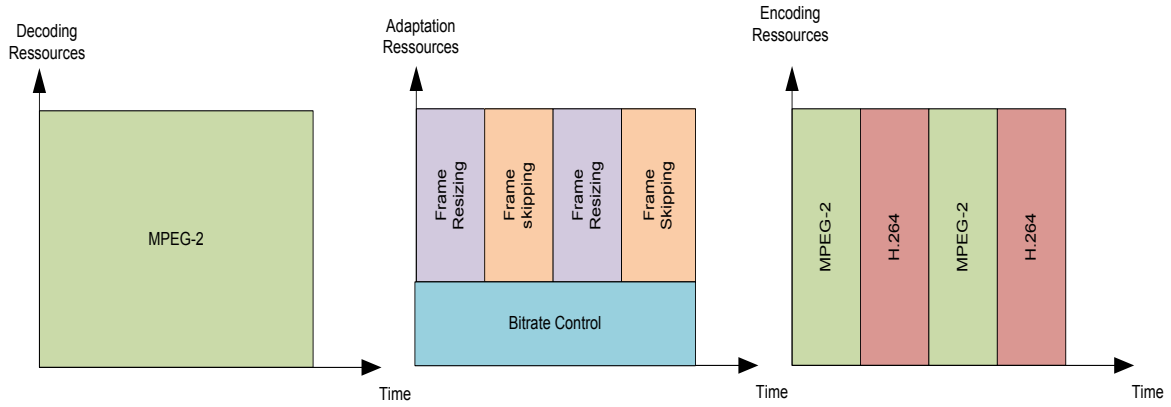


Figure 4- 20 : Temporal Process Switch

The adaptation process operates frame by frame. Each frame is passed with its own parameters. In the aforementioned example, frames from the first use case will carry “adaptation 1”, “no frame skipping”, “half resizing” and “bitrate 200kB”, while the frames from the second use case will carry “adaptation 2”, “1 skip every 10 frames”, “no frame resizing” and “bitrate 300kB”. If a specific adaptation is no needed (e.g. “no frame skipping”), this specific adaptation process will still be performing (e.g. frame skipping) but will result in no change in the stream (e.g. frame skipping of 0 frame). Thanks to this, the right adaptations will be executed on the right stream (1 or 2) and both streams can be interlaced. Adaptation resources consumption over time is depicted in Figure 4- 21.

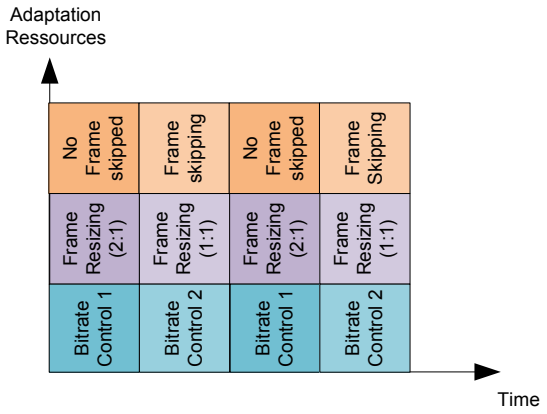


Figure 4- 21 : ARDMAHN static adaptation over time

6 Hardware complexity evaluation

We have implemented the frame resizing process in the unified adaptation format and we have interfaced it with the MPEG-2 adaptation system described in chapter 3. In the following section, design description of the frame resizing process is done and implemented. Implementation results of the system are provided. Bitrate controller module has not been integrated in the system.

6.1 Spatial Downsizing Module

Considering that the unified format uses 4x4 blocks instead of the standard 16x16 blocks, data ordering differs from standard data order. Figure 4- 22 shows the data order at the input/output of the adapter. A 4x4 pixel block is named a µblock. 4 µBlocks form a Y Block (for luminance) or Cb-Cr Block (for chrominance). Following classic video standards, 4 Y Block with the right amount of Cb and

Cr Blocks form a MacroBlock that represent a 16x16 pixel portion of the frame. The amount of Cb and Cr Blocks depends on the chrominance representation (4:4:4, 4:2:2 or 4:2:0 representation). Pixels first describe a µBlock, then µBlocks describe a Y Block and then Y Blocks describe a MacroBlock. Macroblocks come in line to form a Frame.

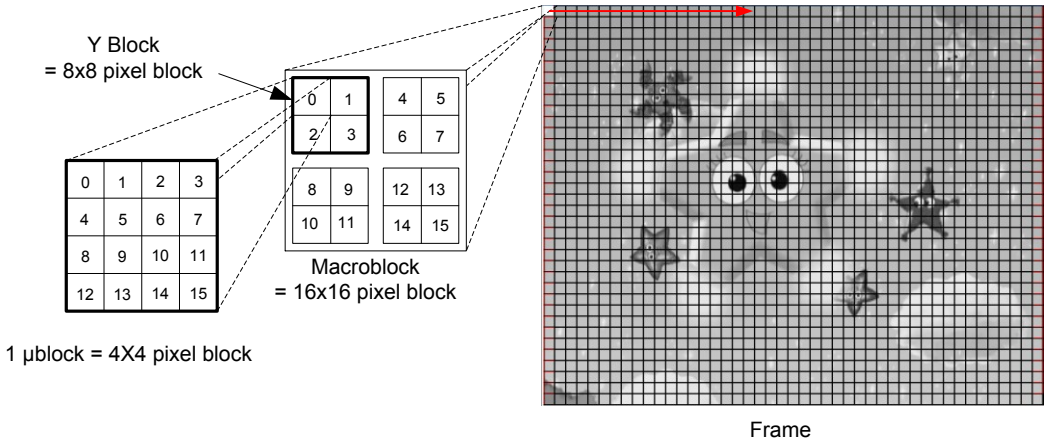


Figure 4- 22 : I/O data order

The spatial downscaling module is responsible for resizing the video frame resolution. It is depicted on Figure 4- 23**.** This process is composed by a *Block Remover*, a *Block Resizer* and a *Block Merger*. The *Block Remover* assures that the number of blocks can be divided by the scale factor. The *Block Resizer* resizes the incoming µBlock by a scale factor of 1, 2 or 4. It operates an average of 1, 4 or 16 pixels to produce 4x4, 2x2 or 1x1 pixel µBlocks. The *Block Merger* forms 4x4 pixel µBlocks from the variable size µBlock, coming from the *Block Resizer*. It processes metadata and sends the results.

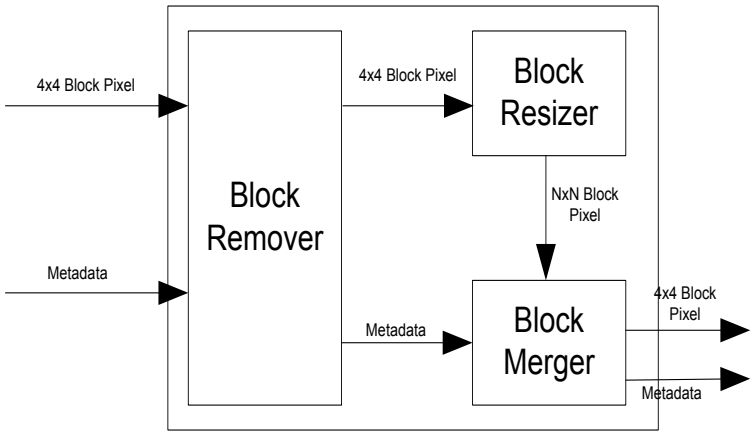


Figure 4- 23 : Frame resizing process

6.2 The "Block Remover" module

A video frame cannot contain a non-integer number of macro-blocks per line or column. The role of the *Block Remover* is to assure, by removing blocks, that there is an integer number of macro-blocks per line and column in the adapted frame. For instance, in a downsizing by 2, the original frame should have an even number of macro-block. If this hypothesis is not fulfilled, the *Block Remover* will remove one macro-block at the end of the line/column.

The Block Remover can be replaced by a “Block Adder” that adds macro-blocks instead of removing them. When adding a macro-block, several issues appear considering the content of the macro-block. We have preferred not tackling this problematic and chose to remove blocks instead of adding them.

6.3 The “Block Resizer” module

The *Block resizer* only operates on pixels that are received in raw order, one value per clock cycle. The pixel process is depicted in Figure 4- 24. A counter controls the routing of data considering the counter value and the scale factor received (1, 2 or 4). For a scale factor of one, the counter adds 0 to the incoming value and outputs the results to operate the identity operation.

For a scale factor of two, the path changes considering if the row is odd or even. For the odd rows, values are summed by pair and stored in the FIFO. For the even rows, the first value is summed to the FIFO value and the second value is summed to the result of the sum. The process repeats itself for the third and fourth data. Then the results are shifted two times and sent.

For a scale factor of four, all 16 data are added - result of the adder is fed back to one of the input of the adder –, shifted four times and sent. Hence the FIFO is used only for the scale factor of two and only stores one data.

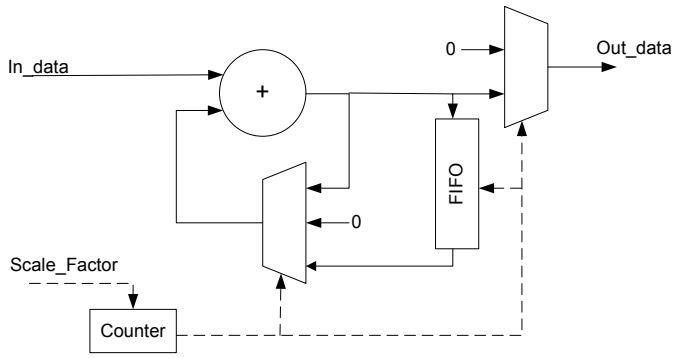


Figure 4- 24 : Block Resizer

6.4 The "Block Merger" module

The *Block Merger* is responsible for:

(1) forming 4x4 µBlocks from the incoming variable size µBlocks;
(2) merging the incoming metadata to form an optimized metadata along the µBlock.

In order to process motion vectors, a design similar to the *Block Resizer* is used to average the incoming horizontal and vertical component of the vector. But to do so, we need to decide if the resulting macroblock will hold motion vector. Indeed, in single picture, neighbors Macroblocks may have different types:

- intra coded (without motion vector);
- inter coded (with only motion vectors).

This decision is bound to the coding type decision that has been depicted in chapter 2. It must be noted that this coding type decision is a priority based decision – i.e. if one of the original Macroblocks is INTRA coded then the resulting Macroblocks is intra coded regardless of other considerations. The selected implementation only requires a hardware comparator and a register. The same type of design goes for quantizer scale decision, where the minimum quantizer scale has been selected among several candidates (one for a scale one resizing, four for a scale two resizing and sixteen for a scale four resizing). The minimum quantizer scale guarantees the best quality possible in the video (even if it rises up the video bitrate).

The *Block Merger* has to form 4x4 pixel µBlock from the incoming NxN pixel µBlock, where N could be 4, 2 or 1, depending on the resizing scale. It is a data re-ordering process that shall output data, as shown on Figure 4- 22 (section 6.1).

When rescaling the frame down by a factor of 2 or 4, the data order is not respected. Figure 4- 25 illustrates this order issue for a scale 2 resizing process. This data disorder comes from µBlocks not containing 4x4 pixels anymore. To reform µBlocks, we used a RAM to store and send data in the right order. The RAM should be designed to support scale 1, 2 and 4 HD frame resizing.

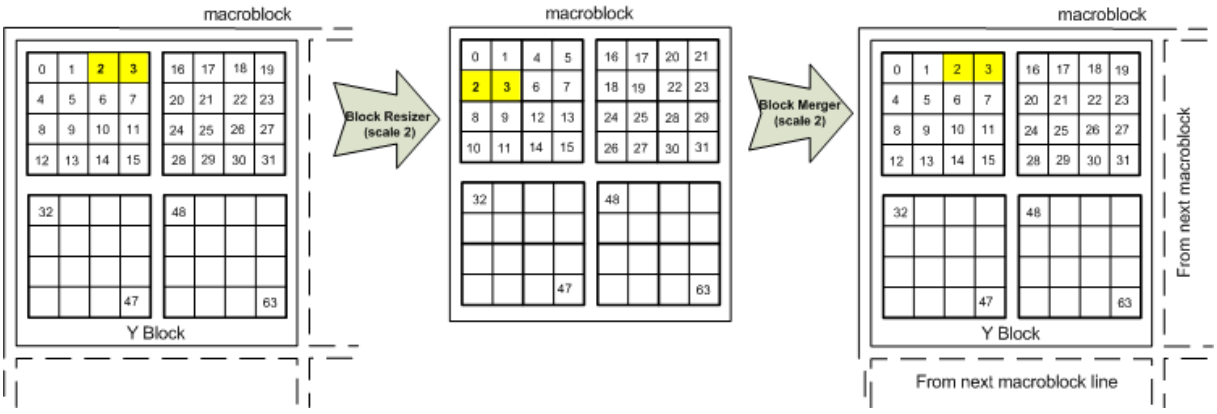


Figure 4- 25 : Data order modification for scale 2 frame resizing

Hence, the RAM should store at least N macroblock lines, where N is the scale factor. The maximum amount of macroblocks in a line is 108, as it is the macroblock amount in 1080p HD Video. However, each macroblock has been pre-processed by the *Block Resizer* and contains MxM luminance data, where M = 16 /N. As a result, the storage amount of pixel the RAM shall have is N x 108 x $(16/N)^2$ = 108 x $16^2/N$, which is maximum with N = 1. An H.264 pixel is coded on 10 bits. The RAM size for luminance is 108x16^2x10=276480 bits = 270 kb. It is augmented by 50%, for a 4:2:0 YCbCr

representation, by 100% for a 4:2:2 representation and by 150% for a 4:4:4 representation. If a particular focus is done on the identity process (scale 1), the RAM size can be reduce by half.

The same system is needed to reorder metadata that are associated to the µBlocks. Figure 4- 26 shows our *Block Merger* system implementation.

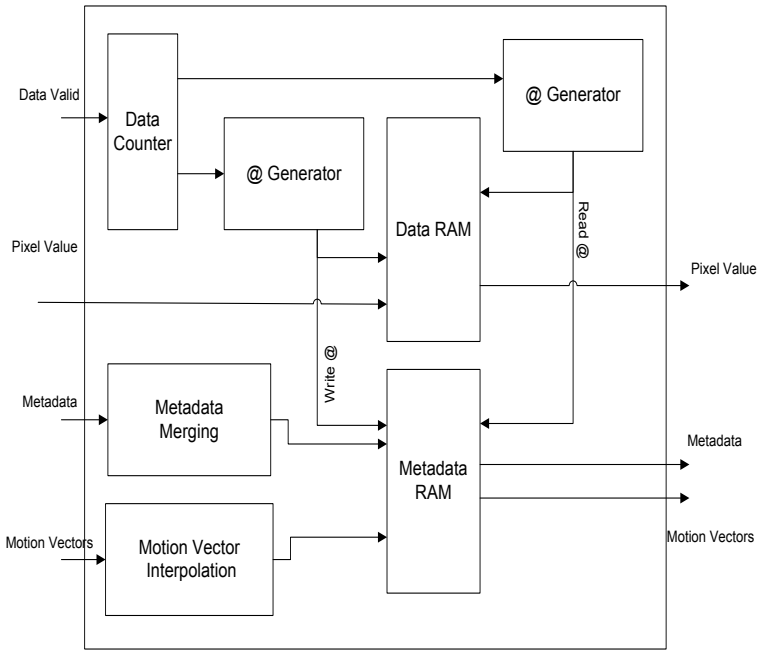


Figure 4- 26 : Block Merger

7 Implementation results

This frame resizing module has been implemented along with our own MPEG-2 design. Implementation results are provided in Table 4- 2. They were obtained using the Xilinx ISE design flow. Targeted FPGA is a Virtex-6 LX240T FPGA that has dynamic reconfiguration capabilities. Those results do not take into account the component mandatory for dynamic reconfiguration such as ICAP and memory to store the partial bitstreams.

We compare our results with designs on the market. To create an adaptation platform with "on the shelves" IPs, using our Adaptation Framework, we need an encoder and decoder designs. The decoding design will have similar FPGA resource consumption to ours as we need a complete decoder. The extra logic required to output metadata is small. Resources consumption of an encoder IP developed by Duma Video Inc [DUMA] is provided in Table 4- 3. These resources are given targeting a Virtex-II FPGA, which is not available in our Tool version. Thus a conversion is needed.

Resources	Adaptation	MPEG-2 (encoding)	MPEG-2 (decoding)	Total
Slices	800	3k	2,6k	7,4k
Multiplier	14	39	14	67
BRAM	10	7	7	24
Freq max (MHz)	200	150	130	130

Table 4- 2 : Design resource consumption

Core	CLBs	BRAM	Multipliers
MPEG-2 encoder	6,5k	73	48

Table 4- 3 : Duma Video inc.'s MPEG-2 encoder resource consumption

To operate such conversion we have taken the Virtex-II and Virtex-6 data sheets. The useful data for our conversion are outlined in Table 4- 4. A Virtex-II CLB is composed of 4 slices and contains up to 128 bits. A Virtex-6 slice is composed of 4 LUT, each of 64 bits. Hence, a Virtex-6 slice contains up to 4*64=256 bits, which is twice as much as a Virtex-II CLB. Our "encoding path" design uses 2,5k Virtex-6 slices, where the MPEG-2 encoder requires 6.5k Virtex-II CLBs which can be roughly converted into 3.25k Virtex-6 slices.

Device Family	CLB	Multipliers	BRAM
Virtex-II	1 CLB = 4 Slices = max 128 bits	18x18 Bit	18 Kb
Virtex-6	1 Slice = 4 LUT + 8 FF; 1 LUT = 64 bits	25x18 Bit	36Kb

Table 4- 4 : Virtex Family Comparison

The Virtex-II BRAM can store up to 18kbits, while the Virtex-6 BRAM can store up to 36 kbits. Thus, a Virtex-6 BRAM stores twice as much data as a Virtex-II BRAM. Our "encoding path" design uses 5 Virtex-6 BRAM, where the MPEG-2 encoder requires 73 Virtex-II CLBs which can be roughly converted into 36 Virtex-6 BRAM.

Results are summarized in Table 4- 5. Our design requires 7% less slice, 80% less BRAMs and 18% les multipliers for the encoding path. Considering that the adaptation and decoding process will use the same resources, the overall gain in our design is: 3% less slices, 54% less BRAM and 11% less multipliers. These results show that our adaptation system is less resources consuming. However, a finer comparison shall be done with IPs designed for the same FPGA in order to get even sharper results. Regarding H.264, the adaptation system actually developed in the ARDMAHN project shows that a higher hardware saving will be provided.

VirteX-6 resources	Duma Video Inc	Our Design	Gain
Slices	3.25k	2.5k	7%
BRAMs	36	5	80%
Multipliers	48	39	18%

Table 4- 5 : Summary of Encoder Gain

Considering performances, our MPEG-2 encoding and decoding path follows the adaptation system defined in chapter 3. The global system (decoding-adaptation-encoding) processes a macroblock in less than 400 clock cycles. Thus, operating at 100MHz, the system adapts a HD video stream (1080p) in real time (0.96s is needed to compute 1s of the video).

The minimal cost of the system - depicted in Table X – is about 7k slices, 67 multipliers and 18 BRAM. Adding H.264 capabilities will only increase the area cost of the reconfigurable zones. However, the adaptation engine costs will stay the same.

8 Conclusion

In this chapter, we have first presented the handling of the codec adaptation problem. Indeed, re-using and combining already developed encoder and decoder to save development costs is a key concept in developing a generic video adaptation system. Both dynamic and static designs have been taken into consideration for our system, according to processing requirements.

Toward design combinations, we have proposed a framework that enables generic adaptation with minimized development costs. It requires a static standardized adaptation path between the decoding and the encoding paths with a unified data format for the adaptation process. This framework is used in the ARDMAHN project, its hardware architecture has been presented as an example.

The unified data format has been created in order to maintain at the maximum the standard features. An analysis has been made and concludes by using the H.264 features as a core structure for metadata communication. The unified format is composed of a 4x4 pixel block that goes along the metadata to manage adaptation the more precisely possible. Examples of format conversion from MPEG-2 and H.264 to the unified format have been done.

Finally, we have presented implementation results in terms of FPGA resources occupation of our system. Results show that compared to the generic adaptation approach (cascading decoder and encoder IPs), the hardware complexity is reduced. Implementation results show a 3% slice saving, a 54% BRAM saving and a 11% multiplier saving, compared to cascading commercialized MPEG-2 encoder and decoder IPs.

Chapter 5 : Novel usages of video adaptation technique

In the previous chapters, we described a system that enables video adaptation on video streamed throughout the network. This system has been implemented in an FPGA based device that is able to instantiate the whole adaptation use case set. This hardware system working in real time performs video stream adaptation according to the network context, i.e. it can be used to reduce the stream bitrate depending on the network usage or to change the video codec according to embedded device capabilities.

In this chapter, we evaluate the opportunity to use this video adaptation framework in order to enable novel usages. Indeed, modifying the video stream in real time permits to enhance the video experience of the End-User, since the adaptation follows the environment variation such as network bandwidth. In addition, it can be used to transform video streams in order to optimize other parameters. More precisely, we will evaluate the opportunity to use video adaptation to reduce the power consumption of the embedded device and measure the video quality impact of the approach. These evaluations aim to demonstrate that novel usages of such an adaptation framework are possible.

1 Study motivation

The work presented in this chapter was motivated by two remarks:

- Embedded video devices have a limited display size. Even if an important part of nowadays-embedded video devices support the decoding of video stream in high definition, the decoded video streams are usually downscaled by the terminal device. This approach is inefficient, as it requires more computation and network usage than required for a well fitted video stream. Furthermore, on small screens, details are less perceivable by the human eyes. Indeed, depending on the terminal screen, HD content is not always perceived as a quality improvement compared to some lower definition content.
- From an end user point of view, the video experience is composed of the video display quality and the avoidance of playback problems. The power consumption of the embedded device can also have an impact on video experience, i.e. if the embedded device consumes too much energy, the end user cannot achieve the playback of the video stream efficiently and the device will quickly run out of energy. In this case, reducing the video stream quality to reduce the power consumption can be advantageous (provided the video quality is not too much impacted).

Concerning this second point, a lot of work has been done on video decoding circuits. All of them [AHM05] and [XIN05] have focused on reducing the video consumption of the processors or of the dedicated circuits in charge of the video stream decoding.

A solution that has not been evaluated consists in modifying the video stream provided to the embedded device to reduce the amount of both the network bandwidth and the amount of decoding computations. Reducing these two parameters directly impacts the energy consumption at the embedded device by reducing the amount of data transfers and the amount of computation executed by the device.

2 Presentation of the different use cases

Modifying the video stream characteristics impacts the video quality and hence the user experience. To evaluate the interest of the proposed approach for real life system, we have considered two distinct use cases.

2.1 First use case: video adaptation according to display characteristic.

The first use case is based on the first remark. In this use case, the video stream received by the embedded device has a larger resolution than the device can play. In order to reduce the computation complexity at the embedded device, we propose to reduce in real time the video dimension beforehand. This video adaptation is possible in real time using the system developed previously. The adaptation system shall not be located on the End-User terminal but on a remote device (Home Gateway, for example). However, modifying the video stream characteristics reduces the video quality due to low-complexity algorithms used to achieve the real time constraint. This impact on video quality must be evaluated in order to quantify the interest of the approach (quality loss versus computation complexity reduction). To realize this evaluation, the experimentation system used is presented in Figure 5- 1.

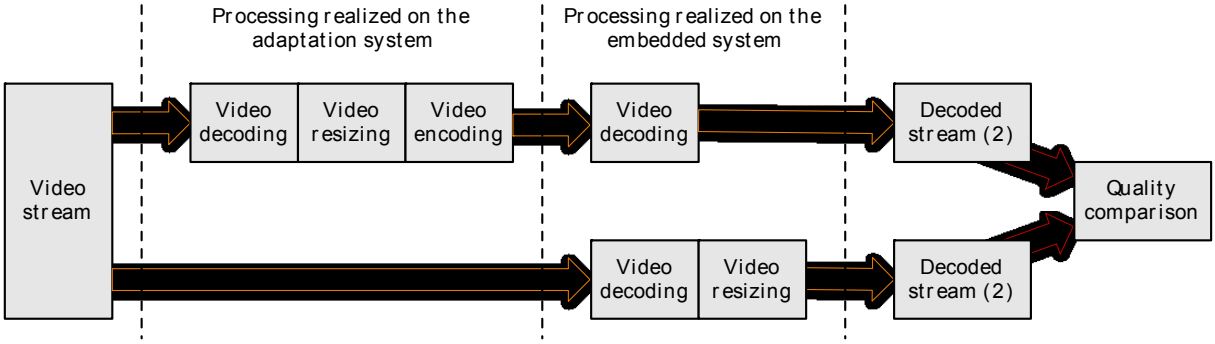


Figure 5- 1 : Experimental approach used to validate the first use case

To fairly estimate the video quality loss, the proposed solution (upper path in the figure) is compared to the video quality obtained using the actual approach (lower path in the figure).

2.2 Second use case: dimension adaptation to reduce power consumption.

The second use case has a different aim. Indeed, the objective is to reduce the video dimension in order to reduce the power consumption of the embedded device. Unlike the first use case, in this approach, the adapted video size is not set to be the optimal size for display – the stream is lowered down. Therefore, the embedded device must upscale the decoded video stream to optimize screen space usage. This operation may introduce more important video quality loss. However, the amount of information to receive and to process from the device point of view is lower compared to the initial stream.

To evaluate the quality results of this approach, the experimentation system described in Figure 5- 2 has been used.

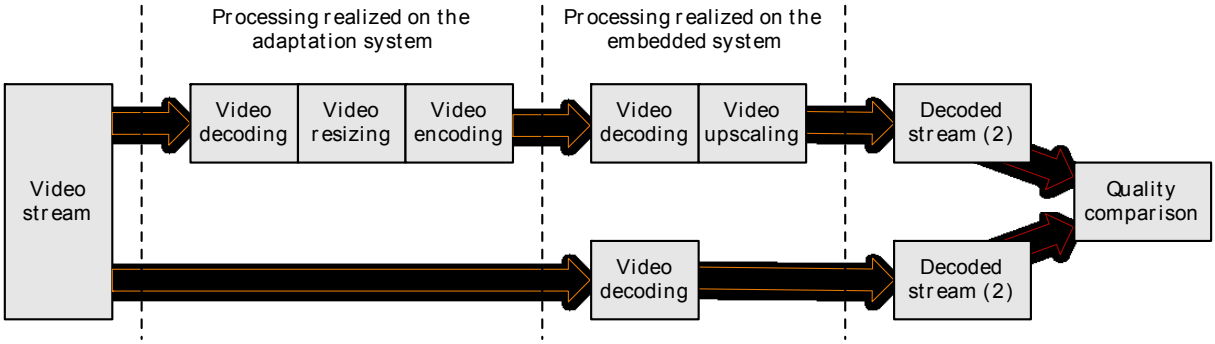


Figure 5- 2 : Experimental approach used to validate the second use case

In this system, the proposed approach first downscales in real time the video dimension. Then, the embedded device decodes this modified video stream. Finally, the decoded stream is upscaled to fulfill the screen dimension. Video information is compared to the traditional approach where the complete video stream is processed by the embedded device.

3 Theoretical evaluation of the proposed approach

The experimentation systems have been executed using the adaptation system described in the section 2.

Videos used by previous experiments (in chapter 3) do not have a sufficiently long duration to allow efficient measurements. In this chapter, the video samples used are trailers of various resolutions. The video set and their characteristics are depicted in Table 5- 1. Each video is encoded in MPEG-2 and MPEG-4 in various resolutions: (1) original, (2) scaled/reduced by a two factor and (3) scaled/reduced by a four factor.

Video Name	Original Resolution	Original Size (bit)		Scale 2 size (bit)		Scale 4 size (bit)	
		MPEG-2	MPEG-4	MPEG-2	MPEG-4	MPEG-2	MPEG-4
300	1920x800	326.9	262.9	95.5	73.5	30.3	24.5
African Cats	1280x720	309.9	266.8	95.3	74.5	26.2	20.6
Winnie the pooh	1280x720	191.2	142.6	59.0	44.3	19.0	14.5
Inception	848x352	44.1	34.5	17.4	14.0	7.6	6.5
Fair Game	1280x544	91.4	66.7	29.2	22.1	11.8	9.0
Kung Fu	848x352	73.3	57.1	25.6	20.4	11.3	9.6
Sucker Punch	1280x544	181.7	137.5	52.7	40.2	20.3	15.9
Avatar	1920x816	221.8	166.9	78.9	63.1	32.5	26.6

Table 5- 1 : Video Characteristics

Videos have been generated using FFmpeg software [FFMPEG]. FFmpeg uses the reference adaptation system (see chapter 2) composed of a fully implemented decoder and encoder at high level high profile that includes an important number of codec features Hence, it achieves good optimization and thus, good bitrate reduction but cannot perform most of the adaptation tasks in real-time with commonly used (in Home Gateways) processors.. Compared to our real-time working hardware system, it has more features and options but similar results both in terms of quality and bitrate reductions are achieved.

It is known that reducing the video spatial resolution also reduces the video bitrate and bitrate reduction permits to reduce the power consumption of the device. Indeed, the lower the bitrate is, the lower the network data transfers are Figure 5- 3 and Figure 5- 4 provide the reduction of the

bitrate achieved by this spatial reduction approach. Figure 5- 3 and Figure 5- 4 show bitrate reduction for both MPEG-2 and MPEG-4 video in the two cases of half downsizing. Figure 5- 3 tackles the downsizing effect from original resolution to half resolution where Figure 5- 4 tackles the downsizing effect from half to quarter resolution.

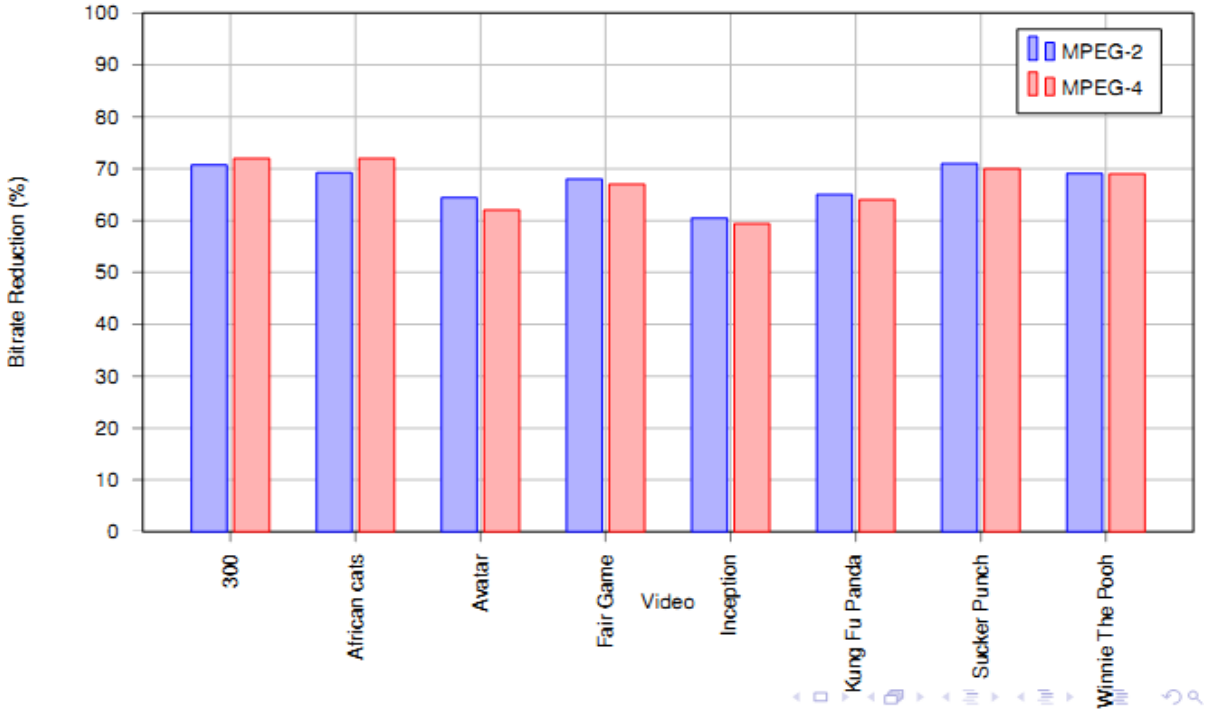


Figure 5- 3 : Comparison of the video stream size from original to half resolution

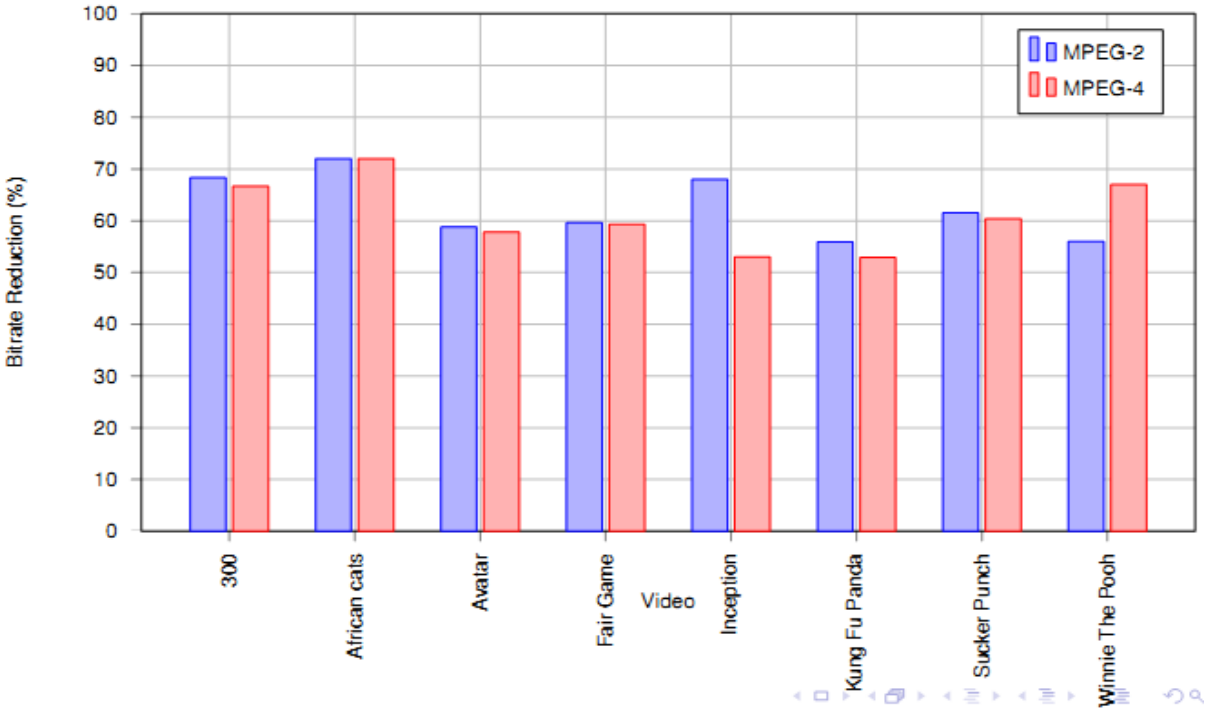


Figure 5- 4 : Comparison of the video stream size from half to quarter resolution

A bitrate diminution from 59% up to 72% is shown in Figure 5- 3 where bitrate diminution is from 53% up to 72% in Figure 5- 4. Those two figures show that a spatial half downscaling induces an average bitrate reduction of 63%. As a consequence, the embedded device data transfers are roughly

reduced by a factor two. This must reduce in theory the power consumption of the network part of the device by two.

However, power reduction is not limited to the reduction of the data transfers. Indeed, reducing the video dimension also reduces the video decoding complexity from the embedded point of view. In order to measure the decoding complexity reduction, the number of macro-blocks that the decoding process has to process to fully decode the video stream has been measured. Figure 5- 5 shows the computation complexity reduction for MPEG-2 video in thousands of macro-block coded.

Experimental results show that the decoding complexity reduction achieved on the embedded device is important. Complexity reduction varies from 73% up to 83%.

In conclusion, we show in this section that preprocessing the video stream to fulfill embedded device characteristics (screen dimension) can efficiently reduce the energy consumption of video streaming. Indeed, preprocessing the video stream reduces first the network activity by 63% in average and secondly the decoding complexity of the video by 78%.

However, video resolution reduction impacts on video quality. The two next sub-sections focus on this quality impact.

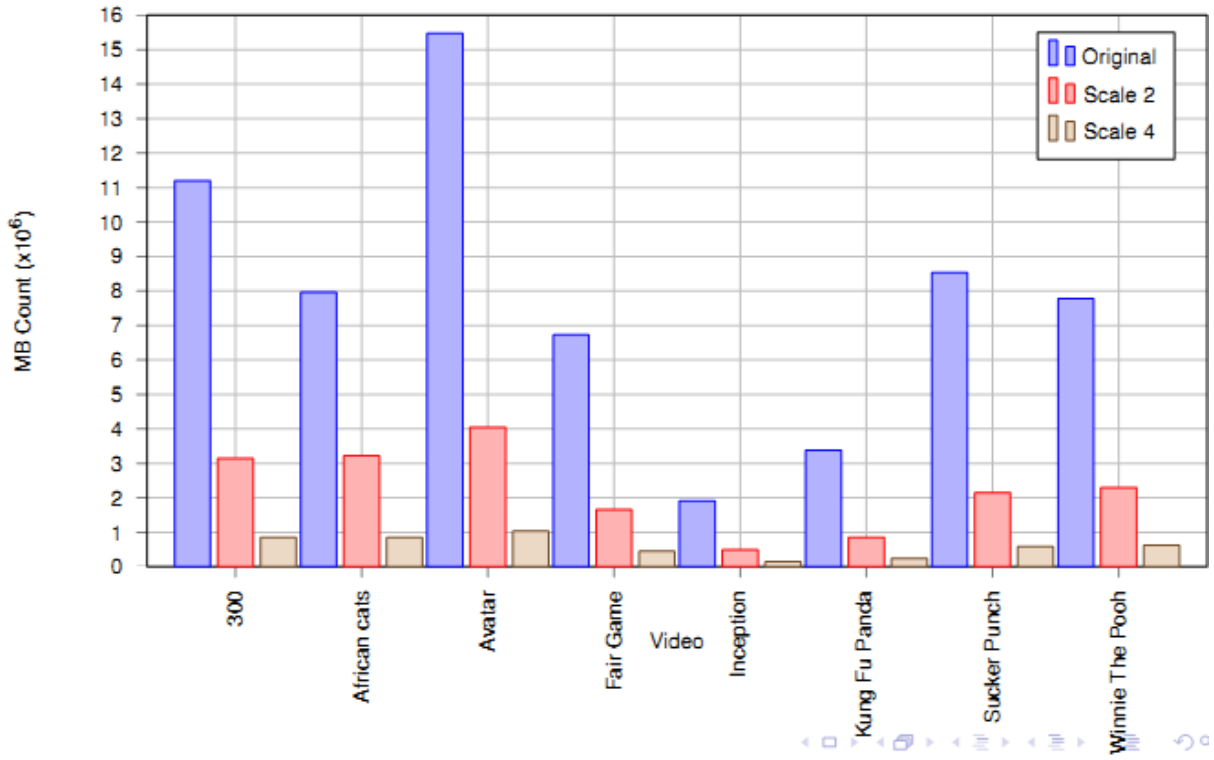


Figure 5- 5 : Comparison of the number of decoded macro-blocks by the embedded device

3.1 First use case: video adaptation according to display characteristic.

The video quality comparisons computed using the first experimental approach are shown on Figure 5- 6 and Figure 5- 7. Figure 5- 6 provides the SSIM [WAN04] information obtained when comparing the video fully decoded by the embedded device to the video preprocessed by the adaptation system. Figure 5- 7 provides an analysis frame by frame of the SSIM information for the video “Fair game” that has the lower SSIM value in Figure 5- 6.

Results provided in Figure 5- 6 show that the quality difference between original video stream and preprocessed one is very small. Indeed, the SSIM metric value is ranged from 95% up to 99%. This mean the video quality impact is unimportant in this use case.

The analysis of the SSIM value frame by frame provided in Figure 5- 7 shows that the low quality impact found using Figure 5- 6 results is achieved for all the video frames. Standard deviation of frame quality is around 2%. This means that there does not exist in the video ugly picture sequence that can generate bad user experience.

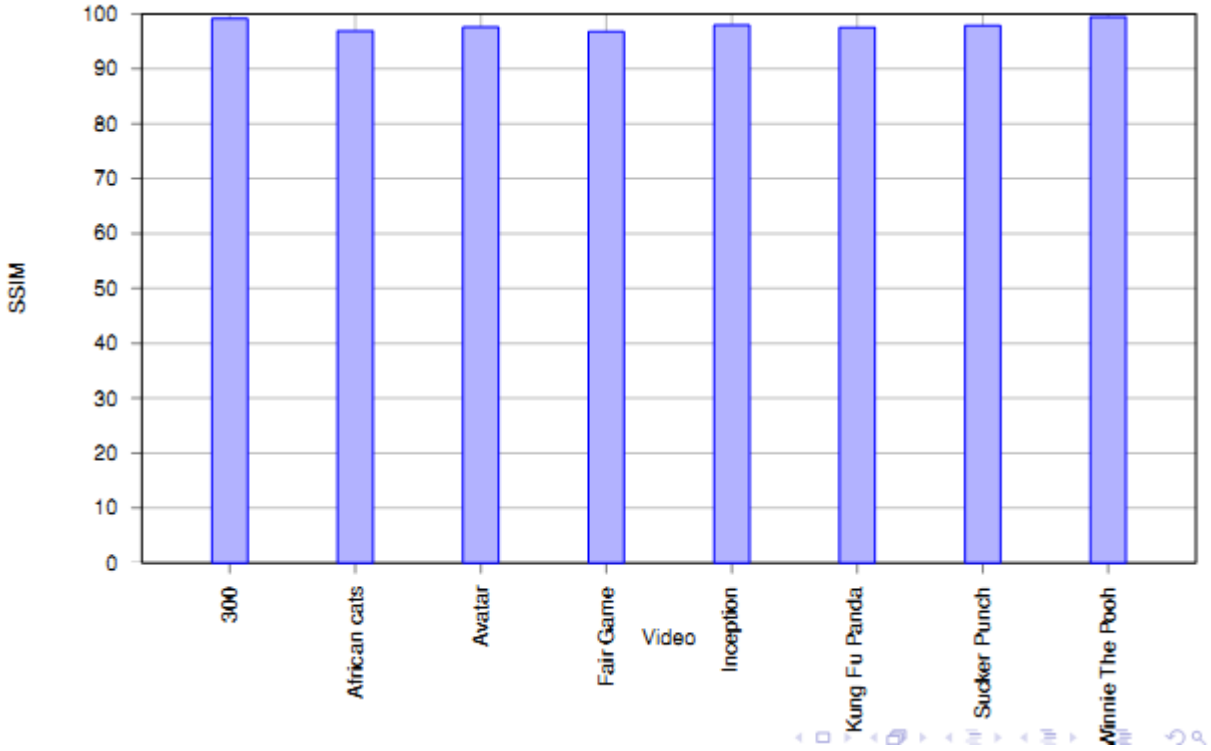


Figure 5- 6 : SSIM quality comparison for various video streams

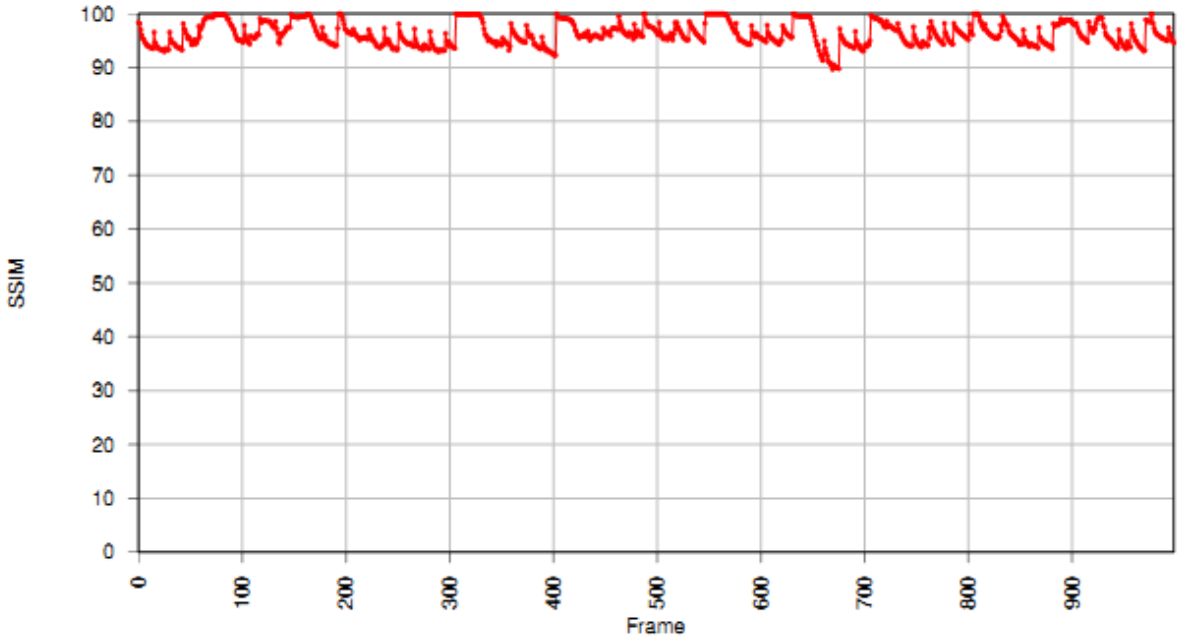


Figure 5- 7 : Frame by frame SSIM comparison for the "Fair game" video stream

For human eye comparison purpose, we provide in Figure 5- 8 and Figure 5- 9, two video frames that have been decoded respectively by the original approach (no adaptation) and by the preprocessed one (adaptation). These frames have been extracted from the video stream “Fair game”. The SSIM values of these two frames correspond to the lowest one of the video sequence.

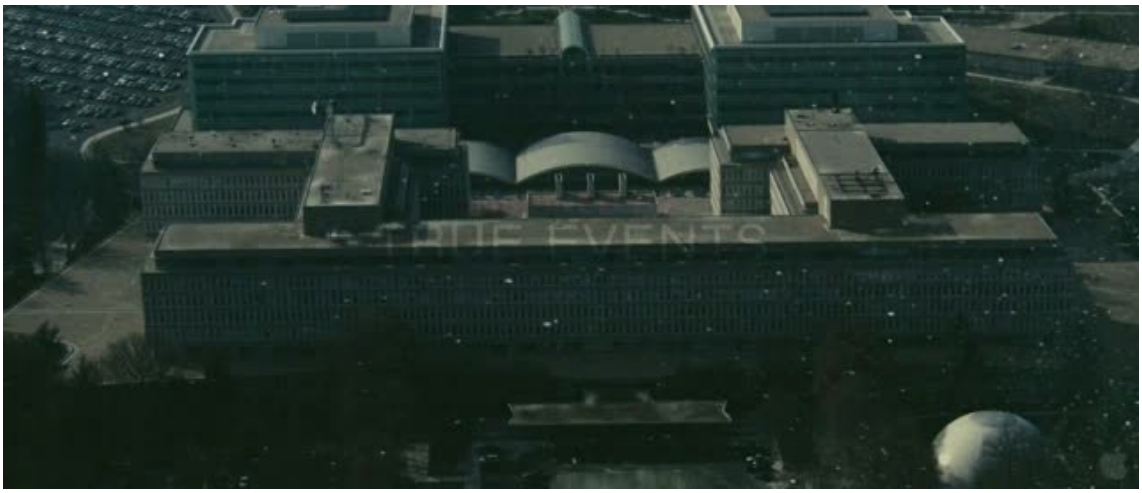


Figure 5- 8 : Frame extrated from the original "Fair Game"

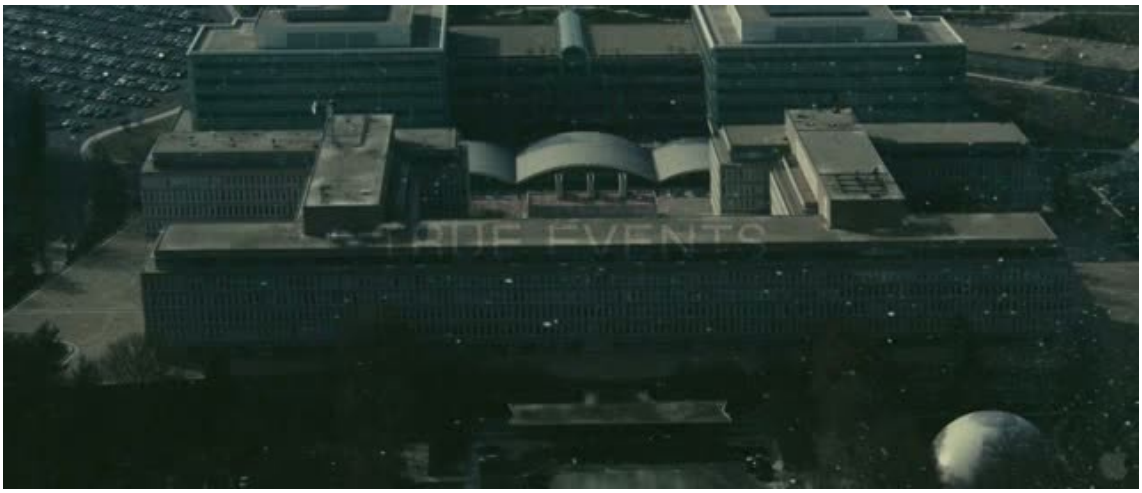


Figure 5- 9 : Frame extrated from the "Fair Game" after processing

As you can see, pictures from Figure 5- 8 and Figure 5- 9 are very similar. It is quite impossible to detect video quality loss when looking at both pictures.

Results provided above show that the video quality is not impacted by the proposed approach.

3.2 Second use case: dimension adaptation to reduce power consumption.

The video quality comparison computed using the second experimental approach is provided in Figure 5- 10 and Figure 5- 13. Figure 5- 10 provides the SSIM information obtained when comparing the video normally decoded by the embedded device to the video preprocessed by the adaptation system and then upscaled by the embedded device. Figure 5- 13 provides an analysis frame by frame of the SSIM information for the video “Kung Fu Panda” that has the lower SSIM value in Figure 5- 10.

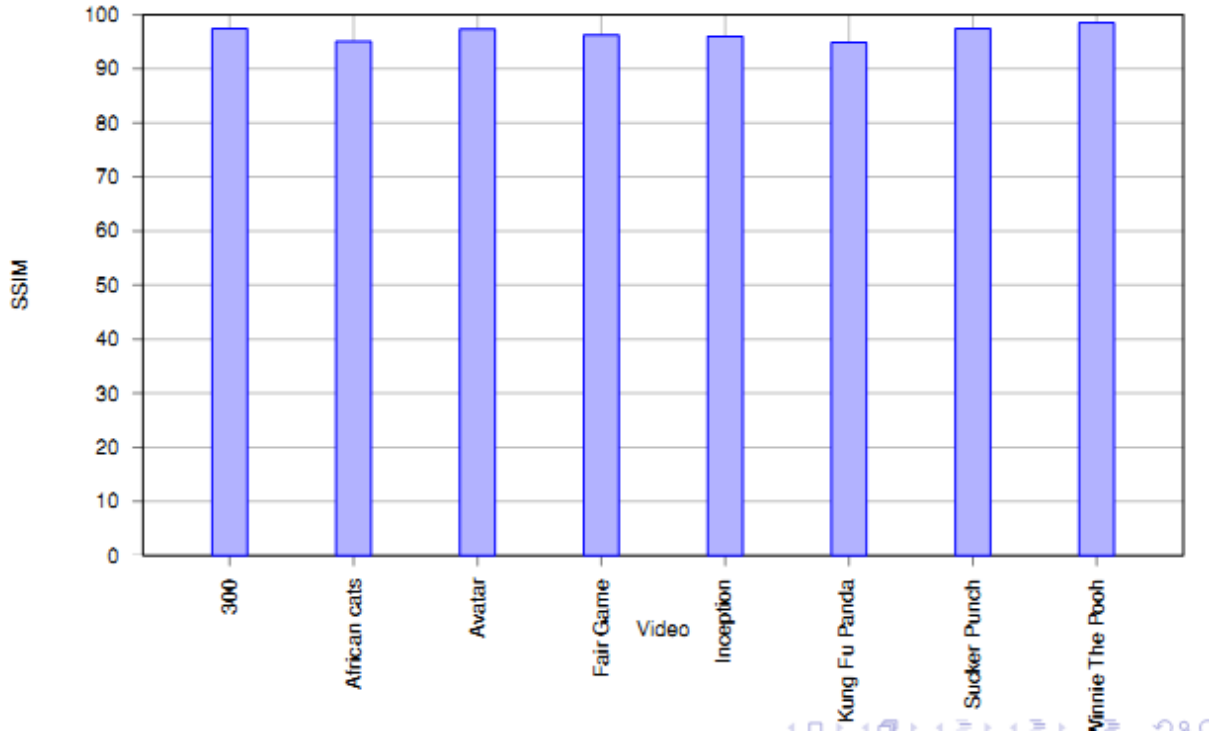


Figure 5- 10 : SSIM quality comparison for various video streams

Results provided in Figure 5- 10 show that the quality difference between the original video stream and the preprocessed one is not obvious. Indeed, the SSIM metric value is ranged from 92% up to 98%. Video quality is lower compared to previous experimentation (use case 1). This acknowledgement can be explained by the fact that in the current experimentation, the video stream needs to be upscaled in order to fulfill the screen dimension. The preprocessed video stream is first downscaled and then re-upscaled. Each of these two inverse transformations impacts on the video quality: the first one removes information from the video stream and the second one tries to regenerate the deleted information.

The video quality decrease is mainly due to the blur effect introduced by the video upscaling process. For human eye comparison purpose, we provide in Figure 5- 11 and Figure 5- 12, two video frames that have been decoded respectively by the original approach and the preprocessed one. These frames have been extracted from video “Kung Fu Panda”

Figure 5- 11 : Frame extrated from the original "Kung Fu Panda"

Figure 5- 12 : Frame extrated from the "Kung Fu Panda" after processing

As you can see, pictures from Figure 5- 11 and Figure 5- 12 are very similar. However, in the second, fewer details appear due to the blurring effect.

The analysis of the SSIM value frame by frame provided in Figure 5- 13 shows that the quality impact found using Figure 5- 10 results is achieved for all the video frames. This means that the picture quality (and the blurring effect) is quite constant during the video sequence.

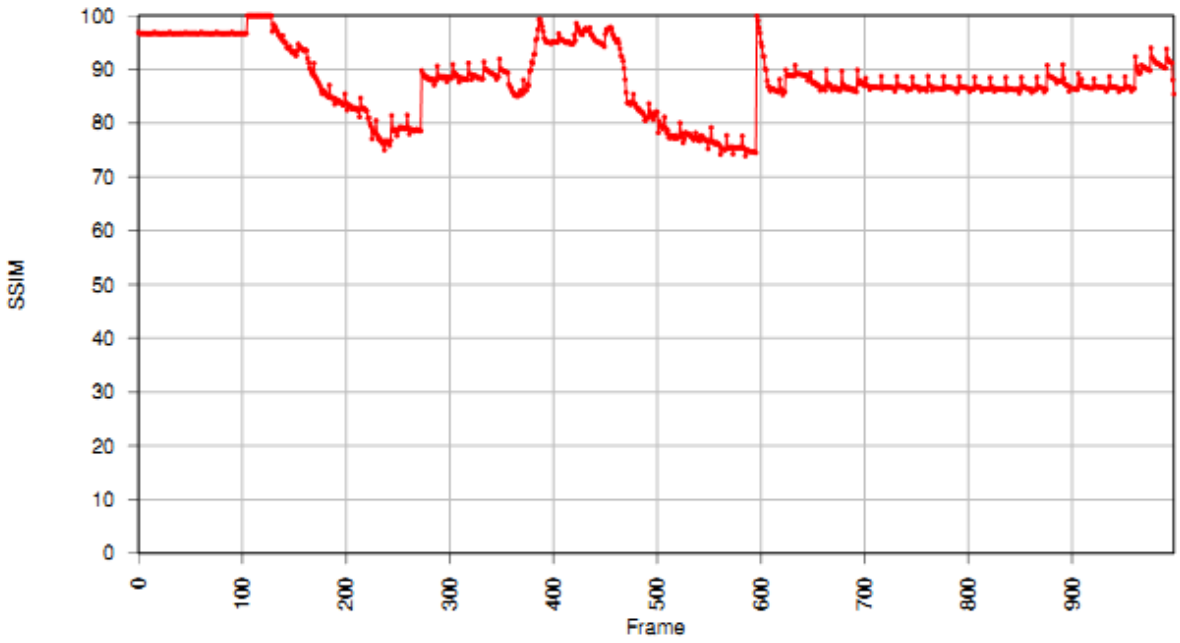


Figure 5- 13 : Frame by frame SSIM comparison for the "Kung Fu Panda" video stream.

Results provided above show that the video quality is impacted by the proposed approach. However, the video sequences are "humanly watchable".

In conclusion, we show in this section that preprocessing the video stream to reduce the power consumption of the embedded device can be achieved. Indeed, preprocessing the video stream reduces first the network activity by 63% in average and secondly the decoding complexity of the video by 78%. All these "savings" are obtained with a very low video quality impact. Quality reduction is quite low even though a blurring effect appears on the picture. Nevertheless, this approach can be stated as efficient in order to enable a longer streaming to an embedded device in case the battery starts to be low (e.g. to finish watching a program) This may increase the overall quality of experience of the service: the video quality may be lower in itself but the service is provided longer (i.e. to the end of the desired program).

4 Experimental evaluations related to the power saving

To evaluate power saving obtained by video resizing, experiments have been held as depicted in Figure 5- 14. A server streams videos to be displayed on a handheld device. Videos are first sent in their original resolution and then in a scaled resolution version (reduced by 2 and then by 4). The handheld device monitors energy consumed by the process.

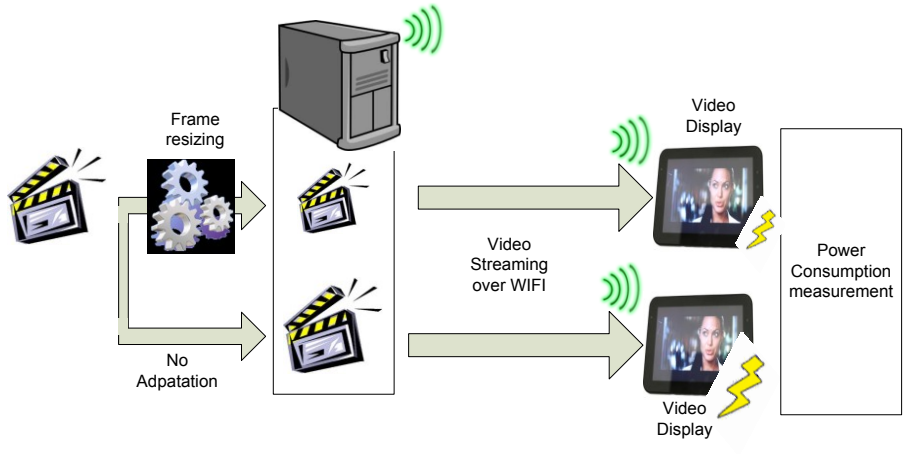


Figure 5- 14 : Experimental Setup

4.1 Experimental details

Two handheld devices are used in order to address both smartphones and tablets. The tablet is a Samsung Galaxy Tab [SAMTAB] which characteristics are shown in Table 5- 2. Screen resolution is 1280x800 pixels which is not high enough to display Full HD video. However, the device supports 1080p video decoding with its 1GHz NVIDIA Tegra™ 2 dual-core processor. The device runs under the Android Honeycomb 3.1 operating system.

Specifications	
Screen Size	7 inches
Screen Resolution	1280x800
Network Capabilities	WiFi 802.11 a/b/g/n, Bluetooth, …
Processor	NVIDIA Tegra 2 dual-core 1GHz
Video Capabilities	1080p

Table 5- 2 : Galaxy Tab Specifications

The smartphone is a Samsung Galaxy S I9000 [SAMPHONE] which characteristics are shown in Table 5- 3. Screen resolution is 480x800 pixels for a 4" screen. The device support 720p video at 30 fps. The device runs under the Android Éclair 2.1 operating system.

Specifications	
Screen Size	4 inches
Screen Resolution	480x800
Network Capabilities	WiFi 802.11 b/g/n, Bluetooth, …
Processor	1GHz ARM "hummingbird"
Video Capabilities	720p

Table 5- 3 : Galaxy S I9000 Specifications

The power monitoring tool used is the PowerTutor application [POWTUT]. This application runs on Android OS and monitors power consumed by each application. Monitored results obtained from hardware monitors take into account hardware usage (Figure 5- 15) such as CPU, LCD display and WIFI.

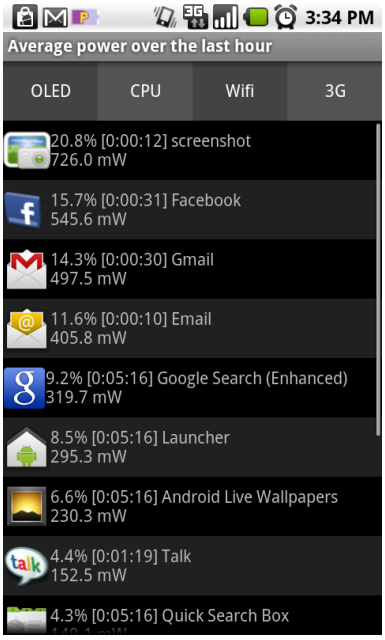


Figure 5- 15 : PowerTutor Screen View

4.2 Experimental results

During the experimental results, video lag and quality decrease have been perceived. Table 5- 4 summarizes the general quality of experience perceived by users.

	Tablet users	Smartphone users
Original Video	Very bad quality of experience (Lag)	Very bad quality of experience (Lag)
Scaled by 2 Video	Good quality of experience	Good quality of experience
Scaled by 4 Video	Average quality of experience (no lag, but perceived decrease in video quality)	Good quality of experience

Table 5- 4 : Perceived Quality of Experience

4.2.1 CPU power consumption

Figure 5- 16 shows mean CPU energy consumption for displaying video in original and downsized version. Both MPEG-2 and MPEG-4 standard are represented as well as both terminals.

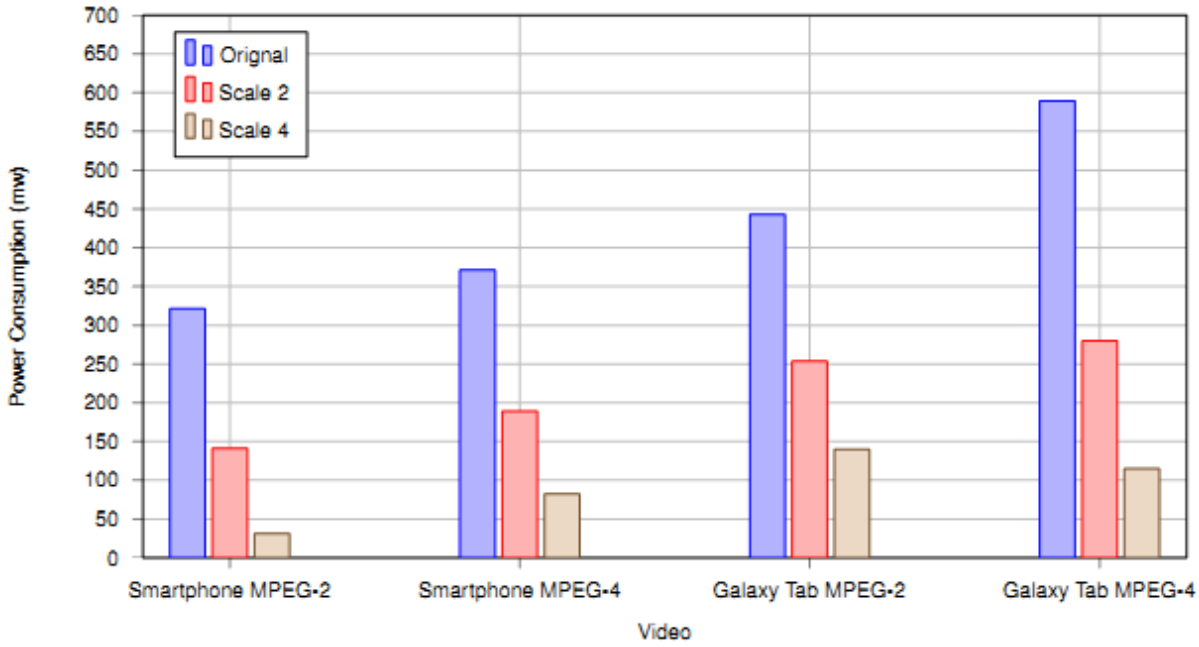


Figure 5- 16 : CPU power Consumption (mW)

Power reduction achieved varies from 42% to 56% for a scale 2 downsizing adaptation (from original to scale 2 version) and from 69% to 90% for a scale 4 downsizing adaptation (from original to scale 4 version).

These results differ from the theoretical 78% power reduction (announced in section 3) for different reasons. First and foremost, our computation complexity reduction was performed considering decoded macroblocs. However, some processing parts are executed whether macroblocs are coded or not. Moreover, energy consumption has a dynamic part (which we try to reduce) and a static part. Hence, we theoretically evaluate that the decoding process would be roughly dynamically 78% less computational but did not consider the static consumption (such as launching the media player). Secondly, CPU power consumption for the media player application takes into account frame resizing at the terminal size to match screen resolution.

4.2.2 WIFI power consumption

Wireless network power consumption during streaming has been monitored and results are shown on Figure 5- 17. Power consumption is impacted at the network interface side.

From original to half resolution downsizing, the smartphone reduces consumption by about 20%. This reduction is very small compared to both the theory and the result achieved by the tablet. We suppose that the causes are correlated to the lag issue when streaming video in their original resolution. This lag may cause not all the data to be transferred and thus the network interface is not triggered as often as it should for video in their original resolution.

For an original to half size resolution at the galaxy tab side, results show a reduction of about 41%. From half to quarter resolution, the downsizing process saves up to 61% of power. Those results are close to the theoretical 63%. The small difference could be explained by the energy needed to start a connection and to assure the quality of the transaction via the use of handshakes and headers.

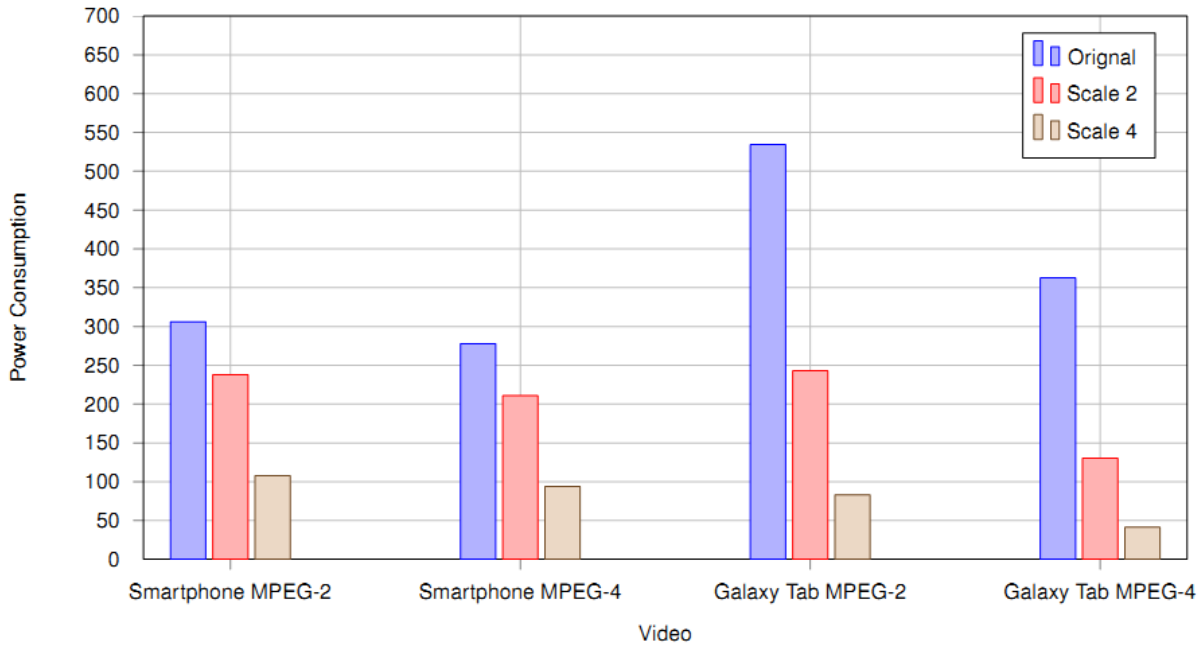


Figure 5- 17 : WIFI power consumption (mW)

5 Recommendations for a generic adaptation system instantiation

We provide below some hints for a novel approach that can be used to reduce the power consumption of embedded devices when displaying video streams. However, these approaches cannot be deployed as is in nowadays systems. In this section, we highlight the modifications required for embedded devices and home networking devices to support video adaptation toward context such as this power saving approach.

Nowadays, home gateways like other network devices are responsible of forwarding data. Hence, they are content agnostic and only rely on networking information to forward network packet or change (if necessary) the network type (ADSL to Ethernet or WIFI or 3G …).

Adding video adaptation features to home gateway requires a whole new environment and functionalities. Deep packet inspection [CAL09] is mandatory for the gateway to be knowledgeable about forwarded content so that video packets can be selected and adapted when needed. Network monitoring tools [STR03] shall be used to understand network state. End-User's terminal capabilities shall be known and linked through an addressable manner. Hence, best-user's experience decision can be done such as adapting the incoming video to the optimal size to avoid lagging or other unpleasant feature.

In order to manage, on the fly, video adaptation, a signaling mechanism shall be used to control the adaptation process such as requiring low resolution video to manage battery state.

The End-User's terminal shall know its own battery state via already embedded battery monitors. However, this battery monitor shall communicate its state to a battery controller that can activate video adaptation remotely to obtain a lower version (and thus less power consuming) of the streamed video. Switching threshold shall be defined considering video duration, power consumption and battery state but can be difficult to evaluate considering some particular videos such as IPTV broadcasting that do not have a proper duration. These additions also consume power and shall be optimized to have the least impact on the battery state.

6 Conclusion

In this chapter, we have proposed a novel approach to raise the quality of experience and to reduce energy consumption at the End-User's terminal side by reducing video frame resolution. We have detailed the two use cases where this reduction can have impact on visual effects. A theoretical evaluation has envisioned a power saving of 78% of the decoding process and 63% of the network interface. According to real experimentations, the estimations reached roughly 50% power saving for the processor and 61% power saving for the network interface.

Such power consumption cannot be integrated as is on nowadays systems. Global system modifications of existing devices, especially on the Home Gateway, are mandatory to trigger video adaptation at the right time to maximize user's experience. A prototype of such a novel Home Gateway is actually in conception at Viotech Communications.

Chapter 6 : Conclusions and future work

1 Summary of key contributions

In this thesis, the need to evolve from a content agnostic network to a more content- and context-aware network has been outlined. This *-awareness feature requires being able to react to content and context information and their potential changes. Towards this, adaptation is the ultimate solution. Achieving adaptation to available network bandwidth and/or terminal features (e.g. screen size, supported codec, …) would significantly bring a step forward to this mandatory evolution in the network domain. The generic video adaptation system proposed within this thesis brings a solution in this direction and is articulated through the support of:

- Bitrate adaptation: the purpose is to adapt the video bitrate to accommodate to the available network bandwidth. Non adapted videos will obviously induce lags effects and errors from packet drops in case of congested networks. On the contrary, an adapted video will permit to avoid effects of congested networks at the cost of a slight and mastered quality loss;
- Codec adaptation: the purpose is to allow any terminal to access any video, whatever the encoding format. Indeed, a video originally encoded in a codec not natively supported by the terminal will result in the inability for the terminal to decode and display the video, leading to a very unsatisfactory End-User experience;
- Spatial resolution adaptation: the purpose is to optimize resource consumption. Indeed, on small screen terminals, high definition video contents will either not be supported or not be noteworthy. In addition, the network will be uselessly overloaded and, at the same time, the terminal will also uselessly consume power to decode and display such video. Adapting the video frame size beforehand would definitely enable an optimized resource consumption of both the network and terminal power.

In order to be efficiently exploited, such a video adaptation platform shall mainly be embedded in low constraint network devices such as home gateways. It must also be noted that to avoid limiting the streaming process and to maintain a real quality of experience to the End-Users, video adaptation must follow real-time constraints.

Video adaptation systems proposed at algorithmic level have been overviewed in order to find a suitable answer to these constraints. In this overview, it has been shown that some already proposed video adaptation systems solve the cost and/or the performance issues but only focuses on one specific video adaptation – i.e. either bitrate reduction or frame rate reduction. Only one costly adaptation system is able to operate the whole set of video adaptation use cases and hence does not satisfy the low cost constraint we have to fullfill. Therefore, to obtain a generic system, we have proposed a solution that achieves a good tradeoff between video quality throughputs and cost/performances.

In order to obtain low cost and performance, a hardware accelerated architecture has been developed to implement the proposed generic adaptation system. However, designing a multi codec video adaptation design is time consuming and, thus, we proposed a three stage framework to reduce development cost by reusing already developed designs. This framework uses an unique adaptation format that allows seamless codec switching for faster design process and improved use of FPGA dynamic reconfiguration feature.

In order to reduce the area cost of the design, we have proposed to use a recently developed FPGA chip that is able to partially reconfigure itself at runtime. Hence, only the needed codecs are setup on the chip at once, mechanism which reduces its size and thus its cost.

Furthermore, indirect effects of the video adaptation system have also been studied. Indeed, resizing video frames induces a reduction of the number of pixels to be decoded and displayed for a frame. Thus, we have proposed to resize video frames in order to reduce energy consumption at the terminal side. By reducing the frame size by half, we have achieved up to 50% power save accompagnied an improvement in Quality of Experience as the video do not lag and/or can be watch through the end before the terminal shutdown due to energy exhaustion.

2 Upgrading the proposed video adaption system and open issues

In this thesis, the core of the video adaptation system has been proposed and developed. However, for a full verification, integrating other codecs to validate the three stage framework is mandatory. The arrival of High Efficiency video coding [HEVC] will open the road to the quest of finding heuristics to translate metadata from known standards (like MPEG-2 or H.264) to newer ones.

Thanks to the video adaptation platform that has been designed in this thesis, we are able to effectively perform adaptation to real videos, in real-time. But there is still a need to know which adaptation to activate according to which situation, in order to be the most efficient possible. This task shall be handled by a decision taking engine. The conception of such engine requires a lot of studies and experimentation and will permit to answer to the questions mentioned below.

When shall we trigger the adaptation?

To decide when adaptation shall be triggerd, some information are required. But what kind of information and how to retrieve it. In chapter 5, we have tackled some key information such as the network state to trigger bitrate adaptation or the terminal capabilities to trigger codec or frame resizing adaptation. In chapter 5, we outlined the need to have a communication protocol that informs the adaptation system that the terminal is running low on battery in order to trigger frame resizing. Towards this, the use of network monitoring tools [STR03] and user profile [AIT11] to support video adaptation shall be studied.

Which adaptation should be performed?

In this thesis, we have seen that frame resizing reduces the video bitrate. This is not the only video adaptation that impacts the bitrate. Reducing the number of frames per second or changing the codec are two other ways to reduce bitrate aside the other bitrate adaptation techniques presented in chapter 2. Studies should be done in order to define the best bitrate reduction technique to perform a given bitrate reduction.

How to optimize resource usage?

As presented in chapter 4, FPGAs are now able to partially reconfigure at runtime. This feature has led us to define an architecture that uses this dynamic reconfiguration for instantiating the desired codec when needed. However, reconfiguration can be used to increase the design performance by switching codecs to perform more than a single adaptation with a single small FPGA.

Chapter 7 : Bibliography references

[AHM05] I. Ahmad, X. Wei, Y. Sun and Y.-Q. Zhang, “Video Transcoding: An Overview of Various Techniques and Research Issues”, IEEE Transactions on Multimedia, vol 7, N°5, October 2005, pages 793-804

[AIT11] Soraya Ait-Chellouche, “Délivrance de services médias suivant le contexte au sein d'environnements hétérogènes pour les réseaux médias du futur”, PhD. Thesis, Université Bordeaux 1, 2011

[ALI] Alicante: media ecosystem deployment through ubiquitous content-aware network environments. European research project within the framework of the EU FP7 in ICT, under grant agreement No. 248652/ /ICT–ALICANTE/, http://www.ict-alicante.eu

[ARD] ARDMAHN: Dynamically Reconfigurable Architecture and Methodology for Self-Adaptation in Home Networking. French research project within the framework of the ARPEGE 2009.

[ASS96] P.A.A. Assuncao and M. Ghanbari, “Post-processing of MPEG-2 coded video for transmission at lower bit rates”, IEEE International Conference on Acoustics, Speech and Signal Proceedings (ICASSP), vol 4,May 1996, pages 1998-2001

[ASS97] P.A.A. Assuncao and M. Ghanbari, “Transcoding of single-layer MPEG video into lower rates”, IEEE Proceedings – Vision, Image and Signal Processing, vol 144, n°6, December 1997, pages 377-383

[ASS98] P.A.A. Assuncao and M. Ghanbari, “Transcoding of single-layer MPEG video into lower rates”, IEEE transaction on Circuits System on Video Technology, vol 8, N°8,December 1998, pages 953-967

[BEL11] K. Belloulata, S. Zhu, J. Tian and X. Shen, “A Novel Cross-Hexagon Search Algorithm For Fast Block Motion Estimation”, 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA),May 2011, pages 1-4

[BJO98] N. Bjork and C. Christopoulos, “Transcoder architectures for video coding”, IEEE International Conference on Acoustics, Speech and Signal Processing, vol 5, May 1998, pages 2813-1816

[BT601] ITU-R “Studio encoding Parameters of Digital Television for Standard 4:3 and Wide Screen 16:9 Aspect Ratio”, Rec. ITU-R BT.601-5 (former CCIR 601), 1982

[CAL09] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, “A survey on Internet traffic identification,” Communications Surveys & Tutorials, IEEE, vol. 11, no. 3, pp. 37–52, August 2009.

[CAR03] J.M.P. Cardoso, “On combining temporal partitioning and sharing of functional units in compilation for reconfigurable architectures”, IEEE Transactions on Computers, October 2003, vol 52, N°10, pages 1362–1375

[CHA07] A.K. Chatterjee, R. Mukherji, S. Tripathi, “An Improved Adaptive Cross Pattern Search (IACPS) Algorithm for Block Motion Estimation in Video Compression” International Conference on Computational intelligence and Multimedia Applications 2007, pages 89-96

[DUMA] www.dumavideo.com

[ELH11] W. Elhamzi, R. Thavot, J. Dubois, J. Gorin, R. Tourki, M. Atri and J. Miteran, “An Efficient Hardware Implementation of Diamond Search Motion Estimation Using CAL Dataflow Language”, International Conference on Microelectronics (ICM), Dec 2011, pages 1-6

[FEA99] N. Feamster and S. Wee, “An MPEG-2 to H.263 Transcoder”, International Symposium on Voice, Video and Data Communications, September 1999

[FFMPEG] www.ffmpeg.org

[FIA11] FIArch Group: Fundamental Limitations of Current Internet and the path to Future Internet (March 2011), http://ec.europa.eu/information_society/activities/foi/docs/current_Internet_limitations_v9.pdf

[H264] Joint Video Team of ISO/IEC MPEG & ITU-T VCEG, “ITU-T Recommendation and international Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)”

[HEVC] B. Bross, W.-J. Han, J.-R. Ohm, G.J. Sullivan, T. Wiegand, “High efficiency video coding (HEVC) text specification draft 8”, document JCTVC-J1003_d7, July 2012

[JAY01] R. Jayaraman, “(When) Will FPGAs kill ASICs?”, Xilinx DAC 2001, https://www.doc.ic.ac.uk/~wl/teachlocal/arch2/killasic.pdf

[KAL05] H. Kalva, B. Petjanski and B. Furht, “Complexity Reduction Tools for MPEG-2 to H.264 Video Transcoding”, Transcation on Information Science and Applications, vol 2, 2005, pages 295-300

[LAV04] M. Lavrentiev and D. Malah, “Transrating of MPEG-2 coded video via requantization with optimal trellis-based DCT coefficients modification”, European Signal Processing Conference (EUSIPCO), September 2004, pages 1963-1966

[LEI02-1] Z. Lei and N.D. Georganas, “Accurate bit allocation and rate control for DCT domain video transcoding”, Canadian Conference on Electrical and Computer Engineering, CCECE, 2002, vol 2, pages 968-973

[LEI02-2] Z. Lei and N.D. Georganas, “H.263 Video Transcoding For Spatial Resolution Downscaling”, International Conference on Information Technology: Coding and Computing, April 2002, pages 425-430

[LIU07] Q. Liu, Q. Hiratsuka, S. Goto and T. Ikenaga, “Two-Steps Cross-Diamond Fast Search Algorithm on Motion Estimation in H.264”, International Conference on Communications, Circuits and Systems (ICCCAS) , July 2007, pages 782-786

[LIU08] T. Liu, “Optimisation par synthèse architecturale des méthodes de partitionnement temporel pour les circuits reconfigurables”. pHD. Thesis, Université Henri Poincaré - Nancy 1, 2008.

[MPEG2] ITU-T Video Coding Expert Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG), “H.262 – ISO/IEC 13818-2:2000 – Information Technology –Generic coding ofmoving pictures and associated audio information: Video”, February 2000.

[POWTUT] http://ziyang.eecs.umich.edu/projects/powertutor/

[PUR99] K. M. G. Purna and D. Bhatia, “Temporal partitioning and scheduling data flow graphs for reconfigurable computers”, IEEE Transactions on Computers, vol 48, June 1999, pages 579–590.

[SCH07] H. Schwarz, D. Marpe and T. Wiegand, “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard” IEEE Transactions on Circuits and Systems for Video Technology, Vol 17 N° 9, pages 1103-1120, Spetember 2007

[SHA00] T. Shanableh and M. Ghanbari, “Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats”, IEEE Transactions on Multimedia, vol 2, N°2, June 2000, pages 101-110

[SHE01] G. Shen, B. Zeng, Y.-Q. Zhang and M.L. Liou, “Transcoder with arbitrarily resizing capability”, IEEE International Symposium on Circuits and Systems, vol 5, 2001, pages 25-28

[SHE97] D. Shen, I.E. Sethi and V. Bhaskaran, “Adaptive motion vector resampling for compressed video downscaling”, International Conference Image Processing, vol 1, October 1997, pages 771-774

[SHE99] B. Shen, I.K. Sethi and B. Vasudev, “Adaptive motion-vector resampling for compressed video downscaling”, IEEE Transactions on Circuits and Systems for Video Technology, vol 9, N°6, September 1999, pages 929-936

[SAMTAB] http://www.samsung.com/fr/consumer/mobile-phones/tablets/tablets/GT-P3110TSAXEF-spec

[SAMPHONE] http://www.samsung.com/fr/consumer/mobile-phones/smartphones/galaxy/GT-I9000HKAXEF-spec

[STR03] J. Strauss, D. Katabi and F. Kaashoek, "A measurement study of available bandwidth estimation", Internet Measurement Conference (IMC'2003), Miami, Florida, USA, October 27-29, 2003.

[SUN03] Y. Sun, X. Wei and I. Ahmad, “Low Delay Rate-Control in Video Transcoding”, IEEE international Symposium on Circuit and Systems (ISCAS) 2003, vol 2, Bangkok, Thailand, May 2003, pages II-660 – II-663

[SUN96] S. Sun, W. Kwok and J.W. Zdepski, “Architecture for MPEG compressed bitstream scaling”, IEEE Transactions on Circuits and Systems for Video Technology, vol 6, N°2, April 1996, pages 191-199

[TAN95] K.H. Tan and M. Ghanbari, “Layered image coding using the DCT pyramid”, IEEE Transactions on Image Processing, vol 4, N°4, April 1995, pages 512-516

[VET02] A. Vetro, T. Hata, N. Kuwahara, H. Kalva and S.-I. Sekiguchi, “Complexity-Quality Analysis of Transcoding Architectures for Reduced Spatial Resolution”, IEEE Transaction on Consumer Electronics, vol 48, N°3, August 2002, pages 515-521

[VET11] A. Vetro, T. Wiegand and G.J. Sullivan, “Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard”, Proceedings of the IEEE, Vol. 99 N° 4, pages 626-642, April 2011

[VUI96] J.E. Vuillemin, P. Bertin, D. Roncin, M. Shand, HH Touati and P. Boucard, “Programmable active memories: Reconfigurable systems come of age”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol 4, March 1996, pages 56–69

[WAN04] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, “Image Quality Assessment: From Error Visibility to Structural Similarity”, IEEE Transactions on image processing, vol 13,N°4, April 2004, pages 600-612

[WAN09] Z. Wang and A.C. Bovik “Mean Squared Error: Love it or Leave it?”, IEEE Signal Processing Magazine, January 2009, pages 98-117

[XIL00] Xilinx: Programmable Logic Book 2000

[XIN02] J. Xin, M.-T. Sun, K. Chun and B.S. Choi, “Motion re-estimation for HDTV to SDTV transcoding”, IEEE International Symposium on Circuits and Systems, vol 4, 2002, pages IV-715 to IV-718

[XIN04] J. Xin, A. Vetro and H. Sun, “Converting DCT Coefficients to H.264/AVC”, Pacific-rim Conference on Multimedia (PCM), 2004

[XIN05] J. Xin, C.-W. Lin and M.-T. Sun, “Digital Video Transcoding”, Proceedings of the IEEE, vol 93 N°1, January 2005, pages 84-97

[YIN00] P. Yin, M. Wu and B. Liu, “Video transcoding by reducing spatial resolution”, International Conference on Image Processing, vol 1, 2000, pages 972-975

[YIN02] P. Yin, A. Vetro, B. Liu and H. sun, “Drift Compensation for Reduced Spatial Resolution Transcoding”, IEEE Transaction on Circuits and Systems for Video Technology, vol 12, N° 11, November 2002, pages 1009-1020

[YU06] L. Yu, J. Li and Y. Shen, “Fast Frame/Field Coding for H.264/AVC” International Conference on Digital Telecommunications (ICDT), August 2006